UNIVERSIDADE DE SÃO PAULO
Escola de Engenharia de São Carlos

Modelagem multivariada da incerteza preditiva de modelos de aprendizado profundo para predição de trajetória de pedestres aplicados a câmeras móveis acopladas a veículos autônomos

Augusto Ribeiro Castro

EESC · USP

Augusto Ribeiro Castro

Modelagem multivariada da incerteza preditiva de modelos de aprendizado profundo para predição de trajetória de pedestres aplicados a câmeras móveis acopladas a veículos autônomos

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, para obtenção do título de Mestre em Ciências - Programa de Pós-Graduação em Engenharia Elétrica.

Área de concentração: Sistemas Dinâmicos

Orientador: Prof. Dr. Valdir Grassi Junior

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

São Carlos

2023

FOLHA DE JULGAMENTO

Candidato: Engenheiro **AUGUSTO RIBEIRO CASTRO**.

Título da dissertação: "Modelagem multivariada da incerteza preditiva de modelos de aprendizado profundo para predição de trajetória de pedestres aplicados a câmeras móveis acopladas a veículos autônomos".

Data da defesa: 15/12/2023.

Comissão Julgadora	Resultado
Prof. Associado Valdir Grassi Junior (Orientador) (Escola de Engenharia de São Carlos/EESC-USP)	Aprovado
Prof. Dr. Fernando Santos Osório (Instituto de Ciências Matemática e de Computação/ICMC-USP)	Aprovado
Prof. Associado Marcelo Becker (Escola de Engenharia de São Carlos/EESC-USP)	Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof Associado **Marcelo Andrade da Costa Vieira**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Carlos De Marqui Junior**

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Ribeiro Castro, Augusto
R923m Modelagem multivariada da incerteza preditiva de modelos de aprendizado profundo para predição de trajetória de pedestres aplicados a câmeras móveis acopladas a veículos autônomos / Augusto Ribeiro Castro; orientador Valdir Grassi Junior. São Carlos, 2023.

Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Sistemas Dinâmicos -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2023.

1. Aprendizado profundo. 2. Estimativa de incerteza. 3. Predição de trajetória de pedestres. 4. Veículos autônomos. 5. Segurança de pedestres. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

Este trabalho é dedicado aos meus pais, que me ensinaram o valor da educação de qualidade e me proporcionaram oportunidades muito melhores das que eles tiveram.

AGRADECIMENTOS

Durante minha vida, tive sempre o suporte dos meus pais, aos quais agradecerei eternamente por terem me proporcionado condições muito melhores do que tiveram. Em especial, agradeço ao meu pai por todo suporte dado durante o mestrado e pelo exemplo de ser alguém que passou por toda a formação acadêmica ao longo da vida, mesmo com filhos e trabalho, e que continua extremamente feliz sempre que pode se envolver com a ciência e com universidades.

Agradeço também ao Prof. Dr. Valdir Grassi Junior pela parceria ao longo de disciplinas de graduação, trabalho de conclusão de curso e orientação do mestrado, além de conselhos sobre a carreira, vida acadêmica e pessoal. Mais do que o papel de mentor acadêmico, o suporte dado contribuiu para minha formação como engenheiro.

É preciso lembrar ainda dos amigos próximos do Laboratório de Sistemas Inteligentes, com quem tive o prazer de conviver presencialmente durante o ano de 2022 e foram minha rede de apoio durante o período. Muito obrigado Eduardo, Kaio, Paulo, Mariana, Nícolas e Nuno pelas discussões acadêmicas e, principalmente, pelas risadas e pela companhia fora do laboratório.

Por fim, agradeço à CAPES pela bolsa de estudos concedida durante 18 meses do mestrado (código 0001, número 88887.601232/2021-00) e à Universidade de São Paulo pela excelência no ensino, pesquisa e infraestrutura que me coloca na posição de privilégio dentro do nosso país. Os recursos físicos do laboratório para treinamento dos modelos de aprendizado profundo com agilidade, o conforto proporcionado aos alunos e a bolsa de pesquisa concebida mesmo em momento de descrença da ciência e deterioração do orçamento foram vitais para a conclusão deste trabalho.

"O conhecimento é uma aventura sem fim à beira da incerteza."

Jacob Bronowski

RESUMO

CASTRO, A. R. **Modelagem multivariada da incerteza preditiva de modelos de aprendizado profundo para predição de trajetória de pedestres aplicados a câmeras móveis acopladas a veículos autônomos**. 2023. 91p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2023.

É esperado que veículos autônomos (VAs) substituam motoristas humanos com expetativa de melhorias de segurança e de operação. Como pedestres são os usuários mais vulneráveis das rodoviais, os veículos autônomos têm o dever de prestar atenção especial a esses agentes para mitigar o número de acidentes de trânsito que os envolvem. A literatura recente em percepção de VAs apresenta métodos focados em predizer o movimento de pedestres tanto antecipando as ações quanto predizendo as futuras trajetórias. Enquanto a primeira pode ser mais fácil de predizer, a última possibilita o VA incorporar conhecimento sobre como o ambiente está prestes a mudar. Essa informação aprimora tarefas vitais para a habilidade de antecipação, como a percepção ativa, planejamento preditivo de trajetória, controle preditivo e interação humano-robô. Entretanto, a aplicação de modelos de aprendizado profundo aumenta a importância de se avaliar a confiabilidade e a eficácia desses modelos antes de utilizá-los na prática, uma vez que as predições obtidas estão sujeitas a ruídos e erros do processo de inferência. Neste trabalho é estudado como treinar um modelo de aprendizado profundo para predizer trajetórias de pedestres e a incerteza preditiva multivariada do modelo com mudanças mínimas na arquitetura. Além disso, são incorporadas condições matemáticas para garantir a estabilidade numérica durante o treinamento. A metodologia proposta aplicada a um modelo baseado em realimentação avaliado no conjunto de dados PIE supera a LSTM Bayesiana, o único modelo neste campo de pesquisa capaz de estimar a própria incerteza das predições. Este trabalho avalia a qualidade da incerteza preditiva obtida nesse experimento para cara trajetória do conjunto de testes utilizando curvas de esparsificação e histogramas bi-dimensionais. A avaliação indica uma correlação forte entre a incerteza preditiva e o erro quadrático médio de cada amostra, garantindo a corretude da metodologia proposta.

Palavras-chave: Aprendizado profundo. Estimativa de incerteza. Predição de trajetória de pedestres. Veículos autônomos. Segurança de pedestres.

ABSTRACT

CASTRO, A. R. **Modeling the Multivariate Predictive Uncertainty in Deep Learning Models Applied to Mobile Cameras Attached to Autonomous Vehicles**. 2023. 91p. Dissertation (Master) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2023.

Autonomous vehicles (AVs) are anticipated to supersede human drivers with an expectation of improved safety and operation. As pedestrians are the most vulnerable road users, autonomous vehicles ought to pay special attention to these agents to mitigate the number of traffic accidents involving them. Recent literature on the perception of AVs has presented deep learning models focused on predicting the movement of pedestrians by either anticipating their actions or predicting their future trajectories. While the former may be easier to predict, the latter enables the AV to incorporate knowledge about how the environment is about to change. This information improves vital tasks to the ability of anticipation, such as active perception, predictive path planning, predictive control, and human-robot interaction. However, the application of deep learning models increases the importance of evaluating the reliability and efficacy of these models before they can be applied in practice since their predictions are subject to noise and inference errors. In this work, we study how to train a deep learning model to predict pedestrian trajectories and the multivariate predictive uncertainty of such models with minimum architectural changes. In addition, we incorporate mathematical conditions to ensure the numerical stability during training. Our experiments using the PIE dataset show how our methodology applied to a feedback-based network outperforms the BayesianLSTM, the only model in this field able to estimate its uncertainty. We evaluate the quality of the predictive uncertainty obtained for each trajectory in the test set using the sparsification plot and bi-dimensional histograms. The evaluation indicates a strong correlation between the predictive uncertainty and the mean squared error of each sample, ensuring the correctness of our methodology.

Keywords: Deep learning. Uncertainty Estimation. Pedestrian Trajectory Prediction. Autonomous Vehicles. Pedestrian Safety.

LISTA DE FIGURAS

Figura 1 – Ilustração de uma rede perceptron multicamadas (MLP) com três camadas neurais. Na ilustração, cada círculo representa um neurônio artificial e cada uma das conexões entre eles são ponderadas por um peso. 31

Figura 2 – Arquitetura de um bloco LSTM padrão. 34

Figura 3 – Ilustração de três diferentes maneiras de associar blocos LSTM para um problema de regressão a partir de dados sequenciais $(X_1, .., X_\tau)$. No primeiro arranjo (a), os blocos LSTM são utilizados para mapear toda a sequência de entrada em um vetor e posteriormente um módulo de regressão realiza a predição dos valores futuros com a dimensão desejada. No segundo arranjo (b), os blocos LSTM são utilizados de forma bidirecional, com uma camada adicionada para processar a sequência de entrada na forma inversa ao final do processamento direto. No terceiro arranjo (c), são ilustrados blocos LSTM na forma bidirecional e empilhados de forma hierárquica. 36

Figura 4 – Modelo unificado de atenção proposto por Niu, Zhong e Yu (2021). . . 37

Figura 5 – Ilustração dos mecanismos de atenção por produto escalar escalado (*scaled dot-product attention*) e por atenção por múltiplas cabeças (*multi-head attention*). 39

Figura 6 – Arquitetura da rede Transformadora proposta por (VASWANI *et al.*, 2017). 40

Figura 7 – Representação esquemática de um VAE: codificador (esquerda) e decodificador (direita). As setas pontilhadas correspondem a processos de amostragem realizados durante o fluxo de informação. 41

Figura 8 – Ilustração do funcionamento de um VAE. A arquitetura aprende um mapeamento estocástico entre um espaço x observado, geralmente de distribuição complicada, e um espaço z, chamado de variável latente, cuja distribuição pode ser simples. 45

Figura 9 – Diagrama esquemático da metodologia proposta por Kendall, Gal e Cipolla (2018) para modelar e utilizar a incerteza aleatória homocedástica como forma de balancear funções de custo de m tarefas de classificação e $N - m$ tarefas de regressão para uma rede neural profunda multitarefas. 47

Figura 10 – Diagrama esquemático correspondente a um *deep ensemble* composto por N modelos treinados de uma rede neural, cada um com um conjunto de pesos θ. 49

Figura 11 – Diagrama esquemático de uma rede neural artificial com *dropout* ativado para a primeira e segunda camadas neurais. Neurônios em azul estão ativados, enquanto neurônios em amarelo estão desativados. 49
Figura 12 – Ilustração da técnica de amostragem Monte Carlo *Dropout*, na qual N amostras de predições são obtidas a partir da técnica de *dropout* durante a predição de um único conjunto de pesos θ da arquitetura. . . 50
Figura 13 – Diagrama esquemático da taxonomia proposta por Gulzar, Muhammad e Muhammad (2021) para a predição de movimento de agentes do trânsito. 51
Figura 14 – Exemplo de detecção de pose humana bidimensional utilizando o OpenPose (Cao *et al.*, 2019). 55
Figura 15 – Diagrama do modelo proposto para predizer trajetórias futuras de pedestres e a matriz de covariância correspondente a cada predição ao longo do tempo. 74
Figura 16 – Histogramas bidimensionais do logaritmo do erro quadrático médio em cada ponto de predição e em função do logaritmo de cada componente de incerteza estimado (considerando apenas os elementos ao longo da diagonal da matriz de covariâncias). As incertezas são bem correlacionadas ao erro já que valores altos do erro levam a valores altos de incerteza (e vice-versa). 79
Figura 17 – Curva de esparsificação do modelo proposto para o conjunto de testes. Ela mostra como o erro quadrático médio varia à medida em que os pontos de incerteza mais alta são removidos de cada trajetória. A curva do oráculo mostra o limite inferior da variação do erro, que corresponde a remover os pontos de erro mais alto em relação à anotação. 80
Figura 18 – Oito exemplos de trajetórias preditas para o conjunto de testes. Da direita para a esquerda, linha a linha: $1^o/15^o/30^o/45^o$ passo predito. As cores dos retângulos representam: vermelho - predição, preto (pontilhado) - valor real, amarelo/verde/azul - predições considerando 1/2/3 desvios padrão para as coordenadas do ponto central 81

LISTA DE TABELAS

Tabela 1 – Descrição das entradas, arquitetura de rede neural utilizada e o conjunto de dados de trabalhos de predição de trajetória de pedestres. 57
Tabela 2 – Tabela de comparação entre os resultados reportados nos estudos de predição de trajetória analisados neste trabalho para os conjuntos de dados PIE e JAAD utilizando $0,5$ s de observação e $0,5/1,0/1,5$ s de horizonte de predição para o erro quadrático médio (MSE) e $0,5$ s para o erro quadrático médio considerando o centro do retângulo que representa o pedestre (C_{MSE}) e o C_{MSE} para o último instante de predição (C_{FDE}). Para métodos multimodais (*), as métricas retratadas dizem respeito à melhor predição dentre 20 dadas como saída. † Modelos que utilizam apenas a trajetória do pedestre como entrada. 66
Tabela 3 – Resultados dos métodos presentes neste texto para o conjunto de dados Stanford *drone*. São mostrados o número de trajetórias preditas (K) e reportadas as métricas de distância média do último ponto da trajetória predita e o valor real (FAD) e distância média entre cada um dos pontos da trajetória predita e a trajetória real (MAD) para a melhor das K trajetórias geradas por cada um dos modelos. 67
Tabela 4 – Resultados para o conjunto de dados ETH+UCY em função de cada uma da cenas do conjunto de dados e também a média geral sobre todo o conjunto. Métricas utilizadas: distância média entre a trajetória predita e a real/distância média entre o destino predito e o real. Resultados para abordagem determinística ou melhor resultado dentre 20 para o caso dos métodos multimodais (*). Para os experimentos, são utilizados 8 quadros de observação e 12 como horizonte de predição. 67
Tabela 5 – Resultados determinísticos e multimodais para o conjunto de dados PIE em pixels. 78
Tabela 6 – Percentual de predições distantes até um dado número de desvios padrões para o conjunto de testes 79

LISTA DE ABREVIATURAS E SIGLAS

AV	*Autonomous Vehicle*
BA-PTP	*Behavior-Aware Pedestrian Trajectory Prediction*
BEV	*Bird's-eye-view*
BiTraP	*Bidirectional Pedestrian Trajectory Prediction*
CCP	Coeficiente de Correlação de Pearson
CNN	*Convolutional Neural Network*
ConvLSTM	*Convolutional Long Short-Term Memory*
CVAE	*Conditional Variational Autoencoder*
ELBO	*Evidence Lower Bound*
FAD	*Final Average Displacement*
FC	*Fully-Connected Layer*
FDE	*Final Displacement Error*
FDP	*Função densidade de probabilidade*
GAN	*Generative Adversarial Network*
GCN	*Graph Convolutional Network*
GMM	*Gaussian Mixture Model*
GPS	*Global Positioning System*
GRU	*Gated Recurrent Unit*
KL	*Kullback-Leibler*
LiDAR	*Light Detection And Ranging*
LSTM	*Long Short-Term Memory*
MAD	*Mean Average Displacement*
MAP	*Maximum a posteriori probability*
MLP	*Multilayer Perceptron*

MSE	*Mean Squared Error*
PCPA	*Pedestrian Crossing Prediction with Attention*
ReLU	*Rectified Linear Unit*
RGB	*Red Green Blue*
RNN	*Recurrent Neural Network*
SGD	*Stochastic Gradient Descent*
SGE	*Stepwise Goal Estimator*
tanh	*Função tangente hiperbólica*
VA	*Veículo autônomo*
VAE	*Variational Autoencoder*

LISTA DE SÍMBOLOS

$\odot$	Multiplicação elemento a elemento entre dois tensores
λ	Autovalor
$\sigma(x)$	Função *sigmoid*, dada por $\sigma(x) = \frac{1}{1+e^{-x}}$
$tanh(x)$	Função tangente hiperbólica, dada por $tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$
σ_i	Desvio padrão da variável i
μ	Média
Σ	Matriz de covariâncias
$cov(a, b)$	Covariância das variáveis hipotéticas a e b
ρ_{ab}	Coeficiente de correlação de Pearson das variáveis hipotéticas a e b, dado por $\frac{cov(a,b)}{\sigma_a \sigma_b}$
$\frown$	Operação de concatenação entre dois vetores
$\mathcal{N}(x; \mu, \sigma^2)$	Distribuição normal da variável x com média μ e desvio padrão σ
I	Matriz identidade
$\hat{L}$	Matriz triangular calculada pela decomposição de Cholesky
$g(.), f(.)$	Exemplo de funções
$i,\ j$	Índices de uma matriz
$i^{(t)}$	Porta de entrada de uma rede LSTM
$o^{(t)}$	Porta de saída de uma rede LSTM
$f^{(t)}$	Porta de esquecimento de uma rede LSTM
W	Matriz de pesos
S, R	Conjuntos de coeficientes preditos que compõem a matriz de covariância
b	Tensor de viés
p	Tensor de pesos de uma célula de memória da rede LSTM
t	Tempo

$\theta,\ \phi,\ \psi$	Parâmetros de modelos
K	Conjunto de chaves das redes transformadoras e matriz de ganho no filtro de Kalman
V	Conjunto de valores
Q	Conjunto de *queries*
d	Dimensão
X	Conjunto de dados
x	Elemento de X
Y	Conjunto de saídas
y	Elemento de Y
z	Variável latente
$p,\ q$	FDPs
$\mathbb{E}$	Esperança
T_h	Horizonte de predição
T_o	Janela de observação
ϵ	Constante de pequeno valor numérico
n	Número de variáveis de entrada
N	Número de amostras
F_{min}	Valor mínimo de número de ponto flutuante representado num dado computador
$\|A\|$, $det(A)$	Determinante de uma matriz hipotética A

SUMÁRIO

1 INTRODUÇÃO 25
1.1 Motivação 25
1.2 Objetivo 26
1.3 Estrutura do texto 26
1.4 Trabalhos publicados 27

2 FUNDAMENTAÇÃO TEÓRICA 29
2.1 Inteligência artificial e aprendizado de máquina 29
2.2 Aprendizado profundo 30
2.2.1 Redes neurais convolucionais 31
2.2.2 Redes *Long Short-Term Memory* (LSTM) 33
2.2.3 Redes Transformadoras 35
2.2.3.1 Mecanismos de atenção 37
2.2.3.2 *Multi-Head Attention* 38
2.2.3.3 Arquitetura original proposta para a rede Transformadora 39
2.2.4 Autocodificadores variacionais (VAE) 40
2.2.4.1 Codificador variacional 41
2.2.4.2 Decodificador variacional 42
2.2.4.3 Processo de treinamento 42
2.3 Estimativa de incerteza no aprendizado profundo 44
2.3.1 Incerteza aleatória heterocedástica 45
2.3.2 Incerteza aleatória homocedástica 46
2.3.3 Incerteza epistêmica heteroscedástica 48
2.3.3.1 *Deep Ensembles* 48
2.3.3.2 Monte Carlo *Dropout* 48
2.4 Considerações finais 50

3 PREDIÇÃO DE MOVIMENTO DE PEDESTRES 51
3.1 Predição de trajetória 52
3.2 Definição do problema 53
3.2.1 Possíveis conjuntos de observações 54
3.2.2 Classes de métodos de solução 58
3.3 Arquiteturas de redes neurais de aprendizado profundo utilizadas . . 58
3.4 Considerações finais 65

4 METODOLOGIA 69

4.1 Estimativa multivariada de incerteza 69
4.1.1 Incerteza aleatória heterocedástica 69
4.1.2 Garantia de estabilidade numérica 70
4.1.2.1 Decomposição de Cholesky 70
4.1.2.2 Modelagem multivariada explícita 71
4.1.3 Incerteza epistêmica heterocedástica 73
4.2 Experimentos 73
4.2.1 Modelos utilizados 74
4.2.2 Métricas utilizadas 75
4.2.3 Configuração do ambiente 75

5 RESULTADOS E DISCUSSÕES 77
5.1 Resultados quantitativos 77
5.2 Validação da incerteza 78
5.3 Resultados qualitativos 79

6 CONCLUSÃO 83

REFERÊNCIAS 85

1 INTRODUÇÃO

Prevê-se que veículos autônomos substituam motoristas humanos com expectativa de melhorias de segurança e de operação do transporte rodoviário (GOUDA *et al.*, 2021). Para isso, algumas tarefas precisam ser cumpridas pelo veículo, dentre as quais a percepção é uma tarefa fundamental que reúne toda a informação necessária sobre o ambiente ao redor do veículo em movimento (JEBAMIKYOUS; KASHEF, 2022).

Uma das tarefas relacionadas à percepção do ambiente é a capacidade de não só identificar os pedestres da cena, mas também de antecipar o comportamento desses agentes por meio do cálculo de trajetórias futuras. A antecipação por meio de uma predição é um requisito para um planejamento seguro e suave da trajetória do veículo em ambientes em constantes mudanças (MANGALAM *et al.*, 2020).

No trânsito, o movimento de pedestres não é linear, é difícil de ser predito analiticamente e pode ser modelado como um evento probabilístico (SANTOS; GRASSI, 2021). Isso se deve ao fato de pedestres não serem como entidades inanimadas que se movem de acordo com as leis de Newton e sim entidades enviesadas que mudam de movimento constantemente e ajustam os seus objetivos enquanto navegam por obstáculos (MANGALAM *et al.*, 2020). Essas mudanças de movimento do pedestre podem ocorrer pela atenção a um veículo em específico ou por estar junto de outras pessoas, dentre outros estados possíveis (SANTOS; GRASSI, 2021).

1.1 Motivação

Com o intuito de abordar o problema de predição de trajetórias futuras de pedestres a fim de antecipar o movimento desses agentes de comportamento tão complexo e reduzir o número de acidentes de trânsito, a literatura aborda uma série de métodos baseados em aprendizado profundo para tratar diferentes características que regem o problema (ALAHI *et al.*, 2016; GIULIARI *et al.*, 2021; SANTOS; GRASSI, 2021; MANGALAM *et al.*, 2020; CHENG *et al.*, 2021; BHATTACHARYYA; FRITZ; SCHIELE, 2018; WANG *et al.*, 2022; RIDEL *et al.*, 2020; RASOULI *et al.*, 2019; YUAN *et al.*, 2021). O aprendizado profundo é uma forma de aprendizagem de máquinas em que a representação dos dados é feita a partir de outras representações mais simples que também são aprendidas durante o processo de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016). Sendo assim, o sistema final apresenta maior capacidade de generalização e mais robustez quando comparado a outras técnicas.

Entretanto, o interesse crescente em modelos de aprendizado profundo aumenta a importância de se avaliar a confiabilidade e a eficácia desses algoritmos antes que eles

possam ser aplicados na prática, já que as predições realizadas estão sujeitas a ruídos e erros de inferência (ABDAR *et al.*, 2021). Dessa forma, este trabalho busca seguir a tendência apontada pela literatura e aborda o problema em questão por meio de aprendizado profundo, mas também utiliza de técnicas para que o modelo treinado possa inferir a incerteza das predições.

Em relação ao estado da arte, apenas a rede LSTM Bayesiana (BHATTACHARYYA; FRITZ; SCHIELE, 2018) modela a incerteza associada ao pesos do modelo, mas tratando as variáveis de entrada como independentes. Isso é feito por meio da própria arquitetura proposta, de modo que não é difícil aplicar técnicas semelhantes a outras redes neurais.

1.2 Objetivo

Este trabalho propõe uma metodologia de treinamento que permite a modelos de aprendizado profundo a capacidade de estimar tanto da incerteza oriunda dos dados quanto dos pesos obtidos para a rede durante o treinamento com a necessitar mínimas mudanças na estrutura da rede. A esse processo de treinamento foram adicionadas restrições importantes para garantir a estabilidade do matemática da função de custo para o caso multivariado sem ter a necessidade de considerar as variáveis de saída como independentes.

Com a metodologia proposta, este trabalho mostra como um modelo simples consegue superar o trabalho de Bhattacharyya, Fritz e Schiele (2018) e ter métricas próximas a de modelos complexos do estado da arte. Além disso, a metodologia se mostra adequada para treinar modelos mais complexos da literatura para predizerem a incerteza da predição sem prejuízos nas métricas obtidas. Por fim, são mostradas discussões a respeito da validade das incertezas obtidas e de como este trabalho é ortogonal aos modelos geradores que dominam o campo de estudo e também têm como saída funções densidade de probabilidade.

1.3 Estrutura do texto

A divisão dos conteúdos abordados ao longo do texto está feita da seguinte forma:

- No Capítulo 2 é feita uma revisão dos conceitos de aprendizado profundo, abordando arquiteturas de redes neurais utilizadas dentro do domínio de predição de trajetória de pedestres e os conceitos envolvidos na estimativa de incerteza de modelos de aprendizado profundo, apresentando os tipos de incerteza modelados neste trabalho;

- No Capítulo 3 os conceitos apresentados no Capítulo 2 são utilizados para compreender os principais trabalhos da literatura que também abordam o tema proposto, bem como uma análise das características de cada método;

- No Capítulo 4 é detalhada a proposta do trabalho para modelar a incerteza preditiva multivariada em modelos de rede neural de aprendizado profundo, adicionando condições necessárias para a estabilidade do sistema quando aplicado à predição de trajetória de pedestres. É descrito ainda o experimento realizado para validar a metodologia, indicando o conjunto de testes utilizado, os modelos treinados, as métricas utilizadas e o procedimento proposto para avaliação da incerteza preditiva;
- No Capítulo 5 são apresentados os resultados obtidos a partir da metodologia proposta no Capítulo 4. É feita uma comparação entre modelos treinados com a metodologia proposta e demais modelos determinísticos e probabilísticos existentes na literatura e são mostrados e discutidos os resultados qualitativos e quantitativos do experimento;
- No Capítulo 6 são apresentadas as conclusões obtidas nesta dissertação em relação à metodologia proposta no Capítulo 4 e aos resultados apresentados no Capítulo 5, destacando a contribuição ao estado da arte.

1.4 Trabalhos publicados

Durante o período de elaboração deste trabalho, foram publicados dois artigos em conferências. Castro., Grassi e Ponti (2022) apresentaram um modelo de rede neural de aprendizado profundo e uma função de custo especial para preencher a informação de profundidade em mapas obtidos por sensores de baixo custo na 17ª edição da *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (VISAPP). Já no tema desta dissertação, Castro e Grassi (2023) apresentaram o conteúdo central deste trabalho na edição de 2023 do Simpósio Latino-Americano de Robótica (LARS).

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos necessários para o entendimento do trabalho e para o desenvolvimento de uma rede neural artificial de aprendizado profundo para solucionar o problema de predição de trajetória de pedestres. Para isso, a Seção 2.1 apresenta o campo de pesquisa em que o método a ser desenvolvido está situado. Depois, a Seção 2.2, apresenta o conceito de aprendizado profundo, discutindo as arquiteturas de redes neurais profundas que se aplicam ao tema deste trabalho. Por fim, a Seção 2.3 introduz os conceitos relacionados à estimativa de incerteza no aprendizado profundo, discutindo técnicas aplicadas durante o treinamento e a avaliação dos modelos.

2.1 Inteligência artificial e aprendizado de máquina

Inteligência artificial pode ser definida como o campo de estudo que busca compreender os princípios que tornam possível um comportamento inteligente de agentes (entidades que interagem como ambiente), de modo que a inteligência é a capacidade de tomar decisões adequadas de acordo com o ambiente, com as limitações e com os objetivos, sendo flexível a mudanças, aprendendo pela experiência e operando com uma quantidade finita de computação (POOLE; MACKWORTH; GOEBEL, 1998).

O aprendizado de máquina corresponde à área da inteligência artificial que busca entender como um computador pode aprender tarefas específicas, como auxiliar no diagnóstico de pacientes doentes, classificar tipos de vinhos, separar materiais por suas características, e outras aplicações (MELLO; PONTI, 2018). Nesse caso, o interesse recai em construir uma função de classificação (ou classificador) $f : X \longrightarrow Y$ tal que para cada vetor de características $x_i \in X$, $f(x_i) = y_i$, com y_i a classe correspondente ao objeto i (MELLO; PONTI, 2018).

Uma das formas de se obter a função de classificador pode ser, por exemplo, criar uma tabela com exemplos descritos por suas características x e as respectivas classes y, obtendo assim os conjuntos X e Y. A partir deles, caso o objetivo seja conhecer a probabilidade de ocorrência dos eventos de Y em função de X, pode ser aplicada, por exemplo, uma regressão logística.

No exemplo acima, o desempenho da função f a ser encontrada depende fortemente das características dos exemplos x que foram escolhidas durante o processo de obtenção do conjunto X (BENGIO; COURVILLE; VINCENT, 2013). Dessa forma, o algoritmo aprende a encontrar a função de classificação a partir de uma representação feita por um especialista.

Para algumas aplicações, entretanto, não é possível definir quais são as caracte-

rísticas que representam um dado, como, por exemplo, no caso de reconhecer carros em imagens (GOODFELLOW; BENGIO; COURVILLE, 2016). Nessas situações, a solução adotada é utilizar técnicas de aprendizado de máquina para aprender não só a função de classificação, mas também aprender a representação dos dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

Esse processo de aprender a forma de representação dos dados para tornar mais fácil a extração de informações úteis ao construir um classificador ou outro tipo de preditor é chamado de aprendizado de representação (BENGIO; COURVILLE; VINCENT, 2013). Uma das formas de aprendizado de representação é o uso de autocodificadores, que são estruturas compostas por dois blocos: o codificador $f_\theta : X \longrightarrow H$, que aprende a representar os dados do conjunto X em um novo espaço H, e o decodificador $g_\theta : H \longrightarrow X$, que reconstrói o conjunto de dados original a partir do espaço H. Ao aprender o conjunto de parâmetros θ que minimiza o erro entre x_i e $g_\theta(f_\theta(x_i)), \forall x_i \in X$, é possível usar apenas o bloco codificador como um método que fornece a representação dos dados (GOODFELLOW; BENGIO; COURVILLE, 2016; BENGIO; COURVILLE; VINCENT, 2013).

O aprendizado profundo, por sua vez, surge a partir do aprendizado de representação, adicionando a capacidade de representar as características a partir de outras mais simples e que também são aprendidas (GOODFELLOW; BENGIO; COURVILLE, 2016). Com isso, se fosse construído um grafo representando as relações que vão sendo obtidas entre as características, o grafo resultante seria composto de diversas camadas, sendo chamado então de profundo (GOODFELLOW; BENGIO; COURVILLE, 2016). Mais informações sobre essa classe de métodos é dada na Seção 2.2, uma vez que esse é um dos principais temas deste trabalho.

2.2 Aprendizado profundo

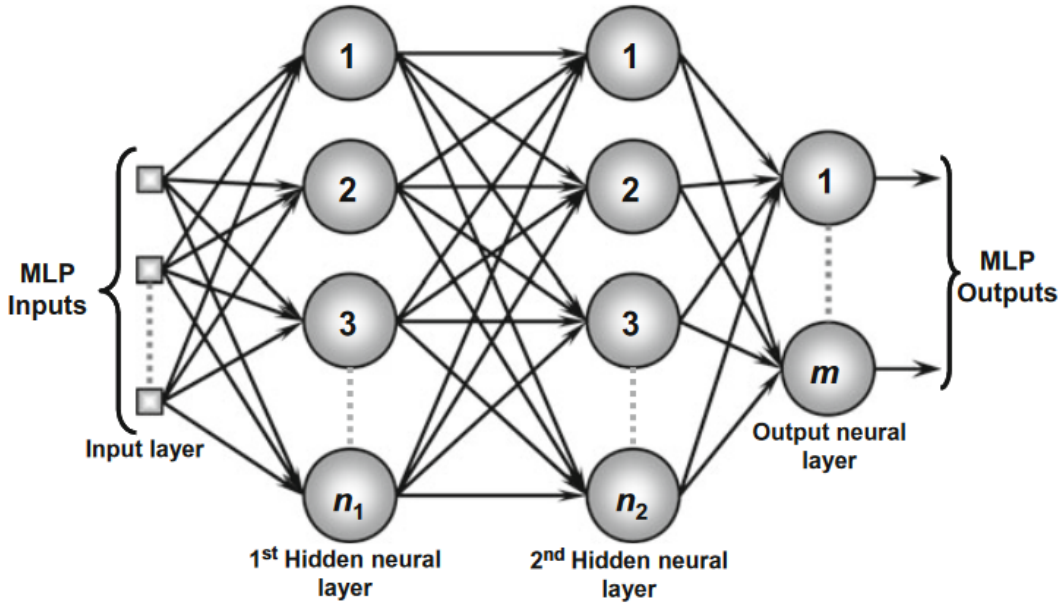
Apesar da grande relevância das redes neurais artificias em trabalhos recentes nos mais diversos temas, como por exemplo processamento de linguagem natural, reconhecimento de fala, interpretação de exames médicos, visão computacional e sistemas de transporte inteligentes (DONG; WANG; ABBAS, 2021), os primeiros modelos matemáticos baseados no neurônio biológico datam das décadas de 40 e 50 (SILVA *et al.*, 2017). Entretanto, após um hiato nos anos 70, a literatura aponta a retomada pelo interesse no uso de redes neurais artificiais a partir de 1985, com a descoberta do algoritmo de retropropagação do erro, que possibilitou o treinamento das redes *Multilayer Perceptron* (MLP) (SILVA *et al.*, 2017; ALOM *et al.*, 2018).

A Figura 1 ilustra um exemplo de rede MLP rasa (de poucas camadas) com três camadas neurais. Essa arquitetura quando há um número relevante de camadas neurais e poder computacional adequado representa o exemplo típico de uma rede de aprendizado

profundo (GOODFELLOW; BENGIO; COURVILLE, 2016).

Apesar de terem perdido espaço para as redes neurais convolucionais em certo ponto, recentemente arquiteturas profundas baseadas nas redes MLP têm ganhado muito destaque em tarefas de visão computacional, como classificação de imagens (YU *et al.*, 2022; LIAN *et al.*, 2021; YU *et al.*, 2021; LIU *et al.*, 2021; TOLSTIKHIN *et al.*, 2021; CHEN *et al.*, 2022), detecção de objetos (LIAN *et al.*, 2021) e em segmentação semântica (LIAN *et al.*, 2021).

Figura 1 – Ilustração de uma rede perceptron multicamadas (MLP) com três camadas neurais. Na ilustração, cada círculo representa um neurônio artificial e cada uma das conexões entre eles são ponderadas por um peso.



Fonte: Figura extraída de Silva *et al.* (2017).

No restante desta subseção, são detalhados os mecanismos básicos de funcionamento das seguintes arquiteturas: redes neurais convolucionais, redes *Long Short-Term Memory*, redes Transformadoras e *Variational Autoencoders*. Os exemplos de aplicação de cada uma dessas arquiteturas no problema de predição de movimento de pedestres são detalhados e discutidos na Seção 3.

2.2.1 Redes neurais convolucionais

Redes neurais convolucionais, do inglês *Convolutional Neural Networks* (CNNs), são um tipo de rede neural projetado para processar informações recebidas na forma de vetores multi-dimensionais, como por exemplo matrizes representando imagens ou espectrograma de áudio, vetores unidimensionais representando sinais no tempo ou entradas

tridimensionais de um vídeo ou informação espacial (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016). O trabalho de LeCun *et al.* (1989) foi o primeiro a empregar CNNs de uma forma compatível com o poder computacional da época e motivou o emprego dessa arquitetura em várias tarefas de reconhecimento, se tornando o estado-da-arte (ALOM *et al.*, 2018).

Um ponto chave no desenvolvimento das CNNs para aplicações de visão computacional foi a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), uma CNN profunda (muitas camadas) e larga (camadas em paralelo) que venceu um desafio de reconhecimento de objetos e motivou o grande interesse por redes neurais de aprendizado profundo (ALOM *et al.*, 2018). Depois, outros modelos vieram com destaque, como a VGGNet (SIMONYAN; ZISSERMAN, 2014), GoogLeNet (SZEGEDY *et al.*, 2015), ResNet (HE *et al.*, 2016) e DenseNet (HUANG *et al.*, 2017).

As CNNs são baseadas na operação de convolução entre uma função $f(x)$ e outra $g(x)$ de modo a obter uma função resultante correspondente à soma dos produtos de $f(x)$ por $g(x)$ ao longo de atrasos da função $g(x)$. Sendo assim, o resultado da operação de convolução $(f * g)(x)$ é dado pela Equação 2.1. Se $g(x)$ for tomada como uma função de pesos, o resultado de $(f*g)(x)$ é uma forma de ponderação do sinal $f(x)$ (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau \tag{2.1}$$

Para o caso de imagens, exemplo mais comum de aplicação das CNNs, os dados são discretos e bidimensionais, de modo que a convolução de uma imagem $I(i, j)$ por um núcleo $K(i, j)$ é dada pela Equação 2.2 (GOODFELLOW; BENGIO; COURVILLE, 2016). Entretanto, a operação descrita pela Equação 2.2 é normalmente trocada pela operação de correlação cruzada (Equação 2.3) para evitar o passo de inversão da imagem.

$$(I * k)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n) \tag{2.2}$$

$$(I * k)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \tag{2.3}$$

A partir das Equações 2.2 e 2.3, diversos filtros foram desenvolvidos para realizar tarefas como suavização da imagem, extração de bordas verticais, extração de bordas horizontais e detecção de cantos (TRUCCO; VERRI, 1998; GONZALEZ; WOODS, 2000). Entretanto, nesses algoritmos para extração de características das imagens, os valores dos pesos dos filtros $K(i, j)$ aplicados sobre a imagem $I(i, j)$ são fixos e definidos pelo projetista, enquanto nas CNNs os valores de cada posição de múltiplos filtros aplicados sobre a imagem são aprendidos durante o treinamento. Dessa forma, o modelo consegue não só

extrair características como cantos da imagem, mas também combinar essas características extraídas ao longo da aplicação de vários filtros, além de ser capaz de aprender a lidar com formas de ruído das entradas.

2.2.2 Redes *Long Short-Term Memory* (LSTM)

Enquanto nas CNNs e nas redes MLPs apresentadas anteriormente a informação flui de forma unidirecional, da entrada da rede até a saída, existem problemas, como a leitura de um texto, em que a compreensão do sentido do texto depende da palavra atual e das palavras lidas até então. Além disso, durante a leitura, não é possível saber quantas palavras serão lidas até o fim do texto, ou seja, o tamanho da entrada e, consequentemente, número de passos a serem executados não são bem definidos. Para essa classe de problemas, que trata do processamento sequencial dos dados, são empregadas redes recorrentes, do inglês *recurrent neural networks* (RNN), nas quais a saída do modelo é utilizada como uma entrada realimentação no próximo passo de predição.

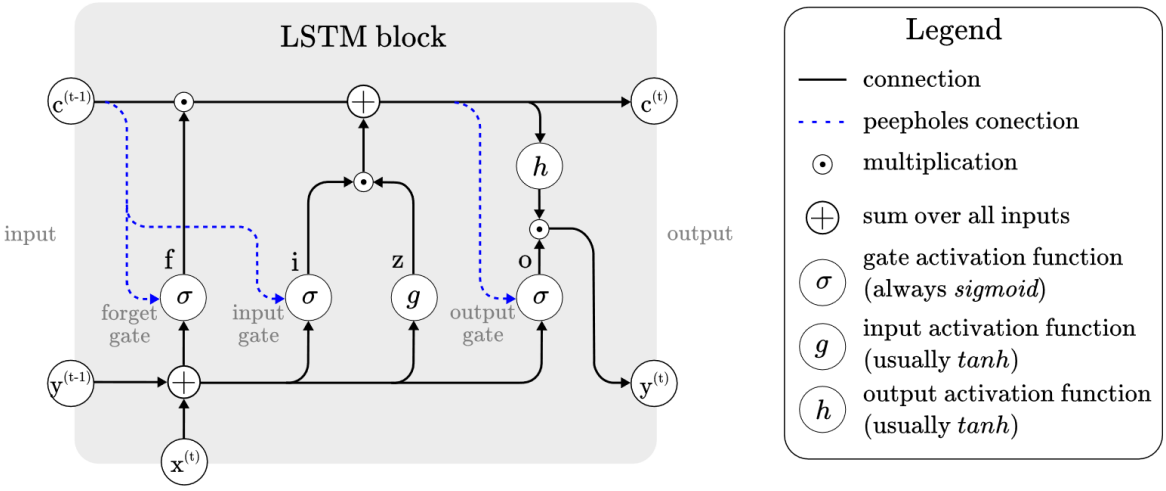
O início do desenvolvimento das RNNs se deu entre 1974 e 1982, com a introdução do conceito de rede neural recorrente e a proposição das redes de Hopfield em 1982 (SILVA *et al.*, 2017; ALOM *et al.*, 2018). Entretanto, apenas com os trabalhos de Hochreiter e Schmidhuber (1997) e de Gers e Schmidhuber (2000), foram propostas as redes longas de memória de curto prazo, do inglês *Long Short-Term Memory* (LSTM), que garantem que os gradientes acumulados ao longo do tempo nem desaparecem nem assumem valores demasiadamente altos (GOODFELLOW; BENGIO; COURVILLE, 2016; HOUDT; MOSQUERA; NÁPOLES, 2020). Com isso, esses modelos possuem não só a capacidade de acumular informações por uma longa duração de tempo, mas também de reiniciar seu estado uma vez que essa informação é utilizada (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outra alternativa ao uso das LSTMs é o uso das *Gated Recurrent Unit* (GRU). Entretanto, como a literatura indica que ambas são de certa forma equivalentes, mas que as LSTM apresentam melhor desempenho quando um volume grande de dados está disponível, já que possuem mais parâmetros que as GRUs (ALOM *et al.*, 2018), esta revisão se atém apenas às redes LSTM.

A arquitetura de uma rede LSTM é composta de um conjunto de sub-redes conectadas de forma recursiva chamadas de blocos (HOUDT; MOSQUERA; NÁPOLES, 2020). Um bloco padrão de uma rede LSTM, como o ilustrado na Figura 2, é composto por uma porta de entrada (*input gate*), uma de saída (*output gate*) e uma de esquecimento (*forget gate*), além de uma célula de memória (*cell*) (HOUDT; MOSQUERA; NÁPOLES, 2020).

A entrada do bloco LSTM na Figura 2, $z^{(t)}$, é composta pela entrada no instante de tempo atual, $x^{(t)}$, e pela saída do bloco do instante de tempo anterior, $y^{(t-1)}$. Para isso,

Figura 2 – Arquitetura de um bloco LSTM padrão.



Fonte: Figura extraída de Houdt, Mosquera e Nápoles (2020).

é utilizada a expressão mostrada na Equação 2.4, em que os pesos W_z e R_z, juntamente com o viés b_z, são associados às entradas e ao resultado. Ao final, é aplicada uma função de ativação g, geralmente a tangente hiperbólica (tanh).

$$z^{(t)} = g(W_z x^{(t)} + R_z y^{(t-1)} + b_z) \tag{2.4}$$

Uma vez determinada a entrada do bloco, é possível então caracterizar o funcionamento das portas de entrada ($i^{(t)}$), de saída ($o^{(t)}$) e de esquecimento ($f^{(t)}$), que regulam o fluxo de informação no bloco, além da célula ($c^{(t)}$) de uma rede LSTM, que guarda valores ao longo do tempo. A porta de entrada combina $x^{(t)}$, $y^{(t-1)}$ e $c^{(t-1)}$ como mostrado na Equação 2.5, com σ a função de ativação sigmoid, W_i, R_i e b_i tensores de pesos das entradas da porta e de viés e $\odot$ a multiplicação elemento a elemento de dois tensores. Além disso, p_i é o tensor de pesos para os dados armazenados na célula de memória. Com $i^{(t)}$, o bloco busca regular quais informações da entrada devem ser armazenadas na célula.

$$i^{(t)} = \sigma(W_i x^{(t)} + R_i y^{(t-1)} + p_i \odot c^{(t-1)} + b_i) \tag{2.5}$$

O contraponto à porta de entrada, que regula a adição de novas informações à memória, é a porta de esquecimento ($f^{(t)}$), que controla quais valores devem ser removidos da célula armazenada até a iteração $t-1$. A expressão de $f^{(t)}$ é mostrada na Equação 2.6 e a definição dos pesos, do viés e da função de ativação é análoga àquela feita para a porta de entrada.

$$f^{(t)} = \sigma(W_f x^{(t)} + R_f y^{(t-1)} + p_f \odot c^{(t-1)} + b_f) \tag{2.6}$$

Uma vez conhecidos $z^{(t)}$, $i^{(t)}$ e $f^{(t)}$, o próximo passo seguindo o fluxo das setas mostrado na Figura 2 é atualizar o valor da célula de memória para o instante atual ($c^{(t)}$), uma vez que a porta de saída também depende do valor de $c^{(t)}$ antecipado (conexão azul). Sendo assim, conforme o diagrama da Figura 2 e das definições acima, o valor atualizado da célula de memória é dado pela Equação 2.7.

$$c^{(t)} = z^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)} \tag{2.7}$$

Resta então definir como calcular a porta de saída do bloco ($o^{(t)}$), que determina como é dada a saída a partir da célula de memória. Dessa forma, de modo análogo às definições feitas nas Equações 2.5 e 2.6, a Equação 2.8 define $o^{(t)}$ em função dos tensores de peso, de viés e da célula.

$$o^{(t)} = \sigma(W_o x^{(t)} + R_o y^{(t-1)} + p_o \odot c^{(t)} + b_o) \tag{2.8}$$

A saída do bloco LSTM para o instante atual ($y^{(t)}$), conforme mostrado na Figura 2, é dada pela Equação 2.9, em que h é uma função de ativação escolhida pelo projetista, mas usualmente igual à tanh.

$$y^{(t)} = h(c^{(t)}) \odot o^{(t)} \tag{2.9}$$

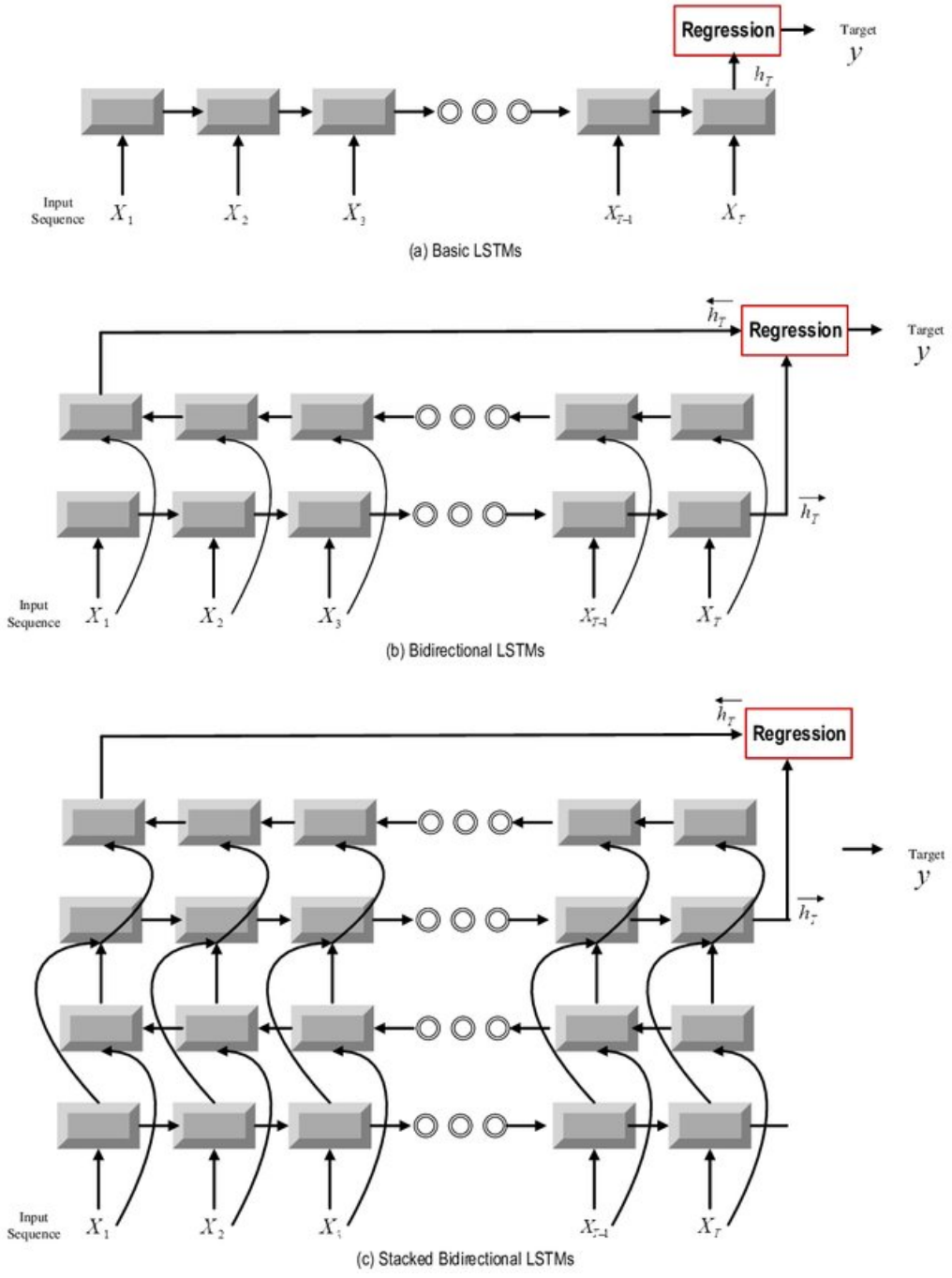
Em aplicações utilizando redes LSTM, entretanto, é possível usar não só um bloco LSTM padrão como o da Figura 2, mas também uma sequência deles empilhados um sobre os outros de forma hierárquica (SUTSKEVER; VINYALS; LE, 2014; HOUDT; MOSQUERA; NÁPOLES, 2020). Ainda, é possível utilizar outros arranjos para os blocos em relação às entradas, como, por exemplo, os arranjos bidirecionais e empilhados mostrados na Figura 3.

2.2.3 Redes Transformadoras

Redes Transformadoras foram propostas por Vaswani *et al.* (2017) como uma alternativa às redes neurais para a tarefa de tradução de sequências, que combinavam recorrência, convoluções, mecanismos de atenção e o emprego de diversos codificadores e decodificadores. Apesar de dependerem de um mecanismo mais simples, as redes Transformadoras se tornaram o estado da arte para a tradução de sequências na data da proposição. Arquiteturas de redes Transformadoras foram utilizadas também em diversas outras tarefas, como visão computacional, processamento de áudio, outros campos de processamento de linguagem natural e até mesmo em áreas da química e da biologia (LIN *et al.*, 2021).

Esses modelos se baseiam inteiramente num mecanismo de atenção para processar dados sequenciais sem depender de nenhuma recorrência (VASWANI *et al.*, 2017). Esses

Figura 3 – Ilustração de três diferentes maneiras de associar blocos LSTM para um problema de regressão a partir de dados sequenciais $(X_1, .., X_\tau)$. No primeiro arranjo (a), os blocos LSTM são utilizados para mapear toda a sequência de entrada em um vetor e posteriormente um módulo de regressão realiza a predição dos valores futuros com a dimensão desejada. No segundo arranjo (b), os blocos LSTM são utilizados de forma bidirecional, com uma camada adicionada para processar a sequência de entrada na forma inversa ao final do processamento direto. No terceiro arranjo (c), são ilustrados blocos LSTM na forma bidirecional e empilhados de forma hierárquica.



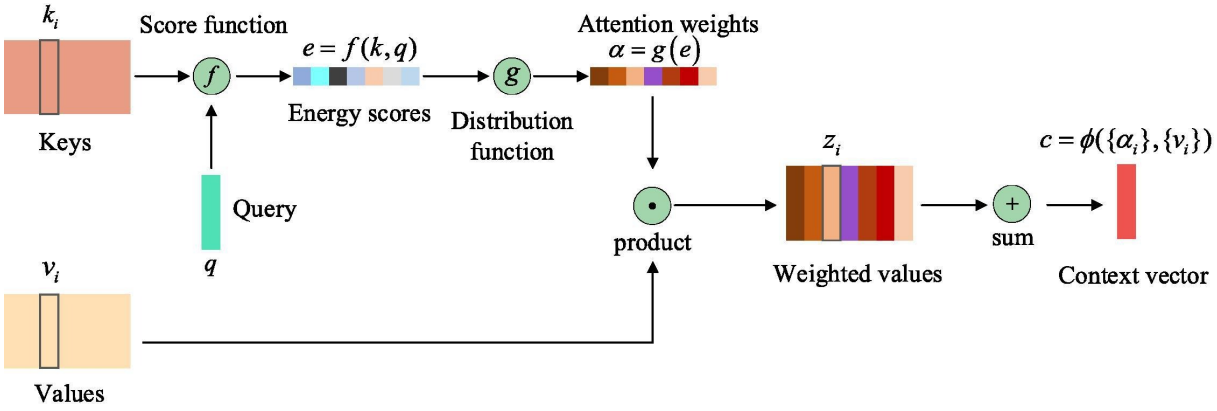
Fonte: Figura extraída de Zhao *et al.* (2017).

mecanismos não são exclusivos dessa arquitetura de rede e podem ser aplicados de diversas outras formas. Niu, Zhong e Yu (2021) apresentam uma revisão da uso de mecanismos de atenção em redes de aprendizado profundo e o estudo serve como base para a breve explicação a respeito desse tema apresentada na subseção 2.2.3.1.

2.2.3.1 Mecanismos de atenção

Bahdanau, Cho e Bengio (2014) foram os primeiros a introduzir o conceito de atenção num modelo de rede neural de aprendizado profundo, que no caso, era aplicado para a tarefa de tradução automática de textos. Depois, vários outros meios de implementar a capacidade de atenção foram propostos. A fim de ilustrar o componente principal presente nos diversos tipos de mecanismos encontrados, Niu, Zhong e Yu (2021) propuseram o modelo unificado de atenção representado na Figura 4.

Figura 4 – Modelo unificado de atenção proposto por Niu, Zhong e Yu (2021).



Fonte: Figura extraída de Niu, Zhong e Yu (2021).

O objetivo de um mecanismo de atenção é calcular uma distribuição de atenção que pondera os valores de entrada do mecanismo (do inglês *values* - V), dando maior importância a algumas entradas em detrimento a outras. A partir disso, é possível calcular então um vetor de contexto, que representa o conjunto de entrada do ponto de vista da atenção empregada.

Para calcular a distribuição de atenção, que representa a importância de cada valor de entrada, uma rede neural primeiramente codifica os valores recebidos, sejam eles uma imagem ou um vetor representando palavras de um texto por exemplo. Essa entrada codificada representa o conjunto de chaves (do inglês *keys* - K). Apesar de, muitas das vezes, K e V corresponderem exatamente a mesma representação do conjunto de entradas do mecanismo de atenção, V é definido como o conjunto em que cada elemento corresponde a um, e somente um, elemento de K.

As chaves, juntamente com um conjunto de questões (*queries*), que são entradas específicas da tarefa sendo abordada (o estado anterior de uma célula LSTM por exemplo),

são utilizadas para calcular uma pontuação de energia (*energy score*), que descreve o quão importante são as questões - q - em relação às chaves para a determinação da saída. A essa pontuação de energia é aplicada uma função de distribuição de ativação, como por exemplo a função $softmax$ ou a função logística, para normalizar e atribuir valores de atenção a cada uma das entradas.

Em resumo, um mecanismo de atenção corresponde a mapear questões e pares de chave-valor a uma saída, em que as saídas, os pares chaves-valor e as questões são todos vetores. A saída é obtida a partir da soma ponderada dos valores, em que o peso associado a cada valor é calculado por uma função de compatibilidade entre a questão e a respetiva chave (VASWANI *et al.*, 2017).

2.2.3.2 *Multi-Head Attention*

O mecanismo de atenção que Vaswani *et al.* (2017) introduziram durante a proposição da rede Transformadora (do inglês *Transformer*) é chamado de atenção de múltiplas cabeças (*multi-head attention*). Nesse modelo, ao invés de utilizar uma única distribuição de atenção para a sequência de entrada, os conjuntos Q (conjunto de todas as *queries*), V e K são projetados em múltiplos e diferentes subespaços a partir de parâmetros treinados e distribuições de atenção são aplicadas em cada um desses subespaços, possibilitando que o modelo considere diferentes representações de subespaços em diferentes posições (VASWANI *et al.*, 2017; NIU; ZHONG; YU, 2021).

A Figura 5 ilustra os mecanismos de atenção por produto escalar escalado (*scaled dot-product attention*) e de *multi-head attention* utilizados pela rede Transformadora (o segundo utiliza o primeiro). A operação executada pelo mecanismo de produto escalar é dado pela seguinte equação, em que d_k é a dimensão dos conjuntos Q e K:

$$c(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{2.10}$$

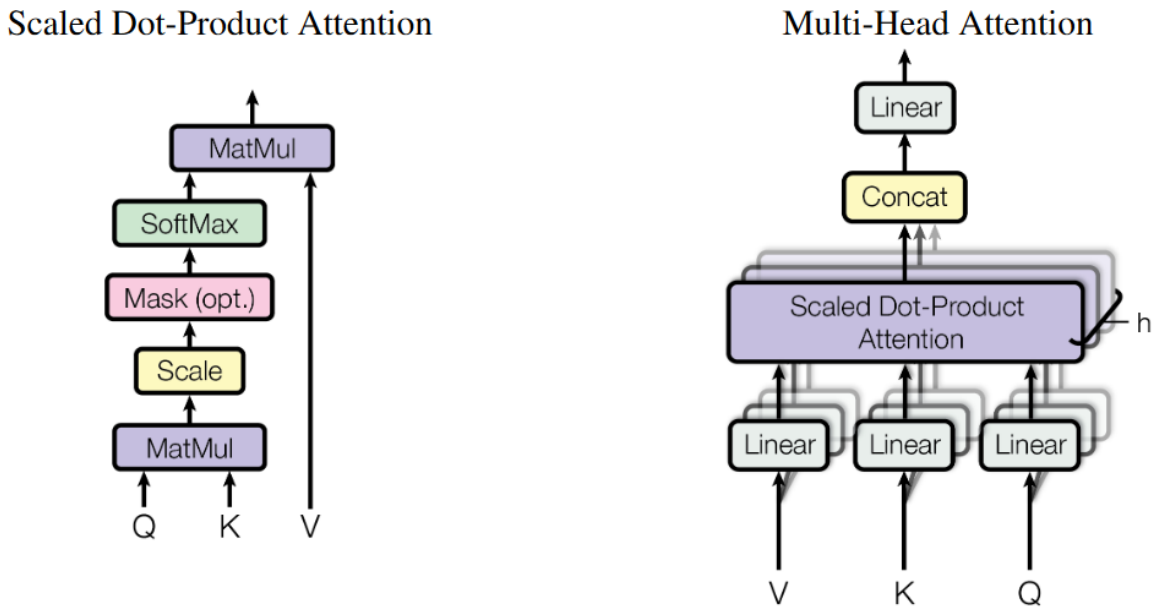
Ao relacionar a Equação 2.10 com o modelo unificado proposto por Niu, Zhong e Yu (2021), a pontuação de energia é dada pela função $f(Q, K)$ descrita por:

$$f(Q, K) = \frac{Q^T K}{\sqrt{d_k}}. \tag{2.11}$$

Seguindo no diagrama mostrado na Figura 4, a função de distribuição de atenção aplicada na Equação 2.10 é a função $softmax$ [1]. Por fim, o produto matricial entre os pesos de atenção e o conjunto de valores corresponde ao passos de produto e de soma que resultam no vetor de contexto.

[1] Para a i-ésima entrada de um vetor z de dimensão k, $softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$

Figura 5 – Ilustração dos mecanismos de atenção por produto escalar escalado (*scaled dot-product attention*) e por atenção por múltiplas cabeças (*multi-head attention*).



Fonte: Figura extraída de Vaswani *et al.* (2017).

Definindo então o mecanismo de *multi-head attention*, primeiro os conjuntos Q, K e V são projetados linearmente para dimensões d_k, d_k e d_v a partir de matrizes de pesos W^Q, W^K e W^V. Esse passo é realizado em paralelo h vezes, de modo que para cada nova projeção, uma nova tupla de pesos W^Q, W^K e W^V é ajustada por treinamento.

Para a i-ésima tupla de projeções, QW_i^Q, KW_i^K e VW_i^V, é aplicado o mecanismo de atenção por produto escalar escalado, resultando no contexto daquela projeção (cp_i) mostrado em:

$$cp_i = c(QW_i^Q, KW_i^K, VW_i^V). \tag{2.12}$$

Os resultados para as h tuplas são concatenados e linearmente projetados para a dimensão de saída (d_{modelo}) desejada a partir de um último vetor de pesos W^O, conforme a equação:

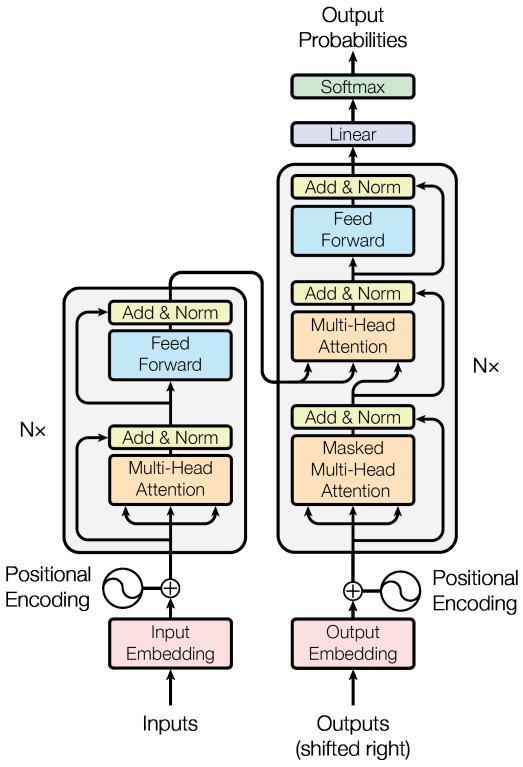
$$MultiHead(Q, K, V) = (cp_1 \frown cp_2 \frown ... \frown cp_h)W^O. \tag{2.13}$$

2.2.3.3 Arquitetura original proposta para a rede Transformadora

A Figura 6 ilustra a arquitetura da rede Transformadora proposta por Vaswani *et al.* (2017). Ela é composta por seis blocos de codificação empilhados e por outros seis blocos de decodificação empilhados.

Os blocos codificador são compostos por dois sub-blocos: uma instância do módulo de atenção proposto seguido por um bloco denso correspondente a uma camada de neurônios da rede MLP. Para cada um desses sub-blocos, são adicionadas ainda conexões residuais, seguidas de uma operação de normalização dos valores obtidos.

Figura 6 – Arquitetura da rede Transformadora proposta por (VASWANI *et al.*, 2017).



Fonte: Figura extraída de Vaswani *et al.* (2017).

Já para os blocos do decodificador, é adicionado um segundo bloco de atenção para processar a saída da pilha de blocos do codificador. Além disso, no primeiro bloco de atenção do decodificador, que recebe as saídas obtidas até o instante anterior ao atual, é adicionado um sistema de máscaras que garante que nenhuma posição futura seja considerada na predição.

2.2.4 Autocodificadores variacionais (VAE)

Os autocodificadores variacionais, do inglês *variational autoencoders* (VAE), foram desenvolvidos de forma independente, simultânea e complementar por Kingma e Welling (2013) e Rezende, Mohamed e Wierstra (2014). Essa arquitetura de rede neural artificial pode ser pensada como uma versão probabilística dos autocodificadores (explicados na Seção 2.1), na qual a saída do decodificador não é um valor de $x_i \in X$, mas sim os parâmetros de uma distribuição de probabilidades de x_i (GIRIN *et al.*, 2020).

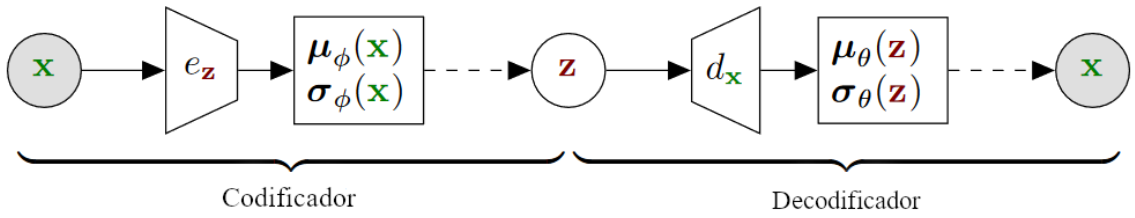
Ao se pensar nos dois módulos que compõem um VAE (codificador e decodificador), é possível identificar que o bloco codificador tem a função de ser um modelo de reconheci-

mento, que mapeia as entradas x para a variável z conhecida como variável latente (uma variável que faz parte do modelo, mas que não é observada por não fazer parte dos dados) (KINGMA; WELLING, 2019).

A Figura 7 ilustra, de forma esquemática, um VAE, mostrando os blocos e as variáveis envolvidas. O codificador, representado por $e_z(x)$, tem como função aproximar $p_\theta(z|x)$ por uma outra função densidade de probabilidade (FDP) $q_\phi(z|x)$ que seja tratável e que possibilite recuperar a variável latente z a partir de x. Para isso, as saídas do bloco codificador, que possui parâmetros ϕ, são a média $\mu_\phi(x)$ e o desvio padrão $\sigma_\phi(x)$ que descrevem a FDP $q_\phi(z|x)$.

Já o decodificador, que implementa a função $d_x(z)$, determina a FDP $p_\theta(x|z)$, a partir de uma rede neural de parâmetros θ, capaz de recuperar uma estimativa de x a partir de z. Mais detalhes sobre cada um desses blocos e sobre o processo de treinamento de um VAE são dados nos subitens a seguir.

Figura 7 – Representação esquemática de um VAE: codificador (esquerda) e decodificador (direita). As setas pontilhadas correspondem a processos de amostragem realizados durante o fluxo de informação.



Fonte: Figura adaptada de Girin *et al.* (2020).

2.2.4.1 Codificador variacional

O objetivo do codificador variacional consiste em determinar a FDP $p_\theta(z|x)$ por meio de uma rede neural artificial de parâmetros θ. Durante o processo de aprendizado, busca-se maximizar a função de verossimilhança marginal (GIRIN *et al.*, 2020):

$$log\ p_\theta(X) = \sum_{x \in X} log\ p_\theta(x). \tag{2.14}$$

Entretanto, por ser modelada por uma rede neural, a probabilidade marginal $p_\theta(x)$ é intratável, ou seja, $p_\theta(x) = \int p_\theta(x, z)dz$ não possui solução analítica ou não pode ser estimada numericamente de forma eficiente (KINGMA; WELLING, 2019; GIRIN *et al.*, 2020).

Dessa forma, a solução apresentada por Kingma e Welling (2013) consistem em introduzir uma distribuição aproximada $q_\phi(z|x)$ implementada pelo codificador de modo

que, a partir do treinamento, o conjunto de parâmetros ϕ otimizado resulte em $q_\phi(z|x) \approx p_\theta(z|x)$. Como, geralmente, a função escolhida para a distribuição aproximada $q_\phi(z|x)$ é a distribuição normal ($\mathcal{N}(x; \mu, \sigma^2)$), a função implementada pelo codificador assume então a forma $e_z(x) = [\mu_\phi(x), \sigma^2_\phi(x)]$.

Com isso, seguindo a notação de Girin *et al.* (2020), é possível calcular $q_\phi(z|x)$ utilizando:

$$q_\phi(z|x) = \prod_{l=1}^{L} q_\phi(z_l|x) = \prod_{l=1}^{L} \mathcal{N}(z_l; \mu_{\phi,l}(x), \sigma^2_{\phi,l}(x)), \tag{2.15}$$

em que o índice l representa a $l-$ésima entrada de um vetor e L é a dimensão escolhida para a variável latente.

2.2.4.2 Decodificador variacional

Conforme citado anteriormente e ilustrado na Figura 7, o decodificador variacional tem como função recuperar x a partir de z. Dessa forma, com o uso dos parâmetros do decodificador, é possível calcular a probabilidade de intersecção de x e z ($p(x, z)$) utilizando:

$$p(x, z) = p_\theta(x|z)p(z), \ para \ p(z) = \mathcal{N}(z; \mathbf{0}, \mathbf{I}). \tag{2.16}$$

Além disso, é possível recuperar a probabilidade marginal de x ($p_\theta(x)$) a partir de:

$$p_\theta(x) = \int p_\theta(x|z)p(z)dz. \tag{2.17}$$

Tomando, então, $d_x(z) = [\mu_\theta(z), \sigma^2_\theta(z)]$ como a função implementada pela rede neural responsável pelo decodificador considerando $p_\theta(x|z)$ dado pela distribuição normal, como exemplificado por Kingma e Welling (2013) - o desenvolvimento é independente da escolha da FDP e Girin *et al.* (2020) citam outras FDPs dependendo do escopo do problema - a seguinte equação mostra como calcular $p_\theta(x|z)$ a partir da saída do decodificador, e do vetor de entradas x de dimensão F:

$$p_\theta(x|z) = \prod_{f=1}^{F} p_\theta(x_f|z) = \prod_{f=1}^{F} \mathcal{N}(z_f; \mu_{\theta,f}(z), \sigma^2_{\theta,f}(z)). \tag{2.18}$$

2.2.4.3 Processo de treinamento

Kingma e Welling (2013) mostraram que, apesar de $p_\theta(X)$ ser intratável para o caso em que é utilizada uma rede neural artificial, conforme mencionado anteriormente, é possível calcular um limite inferior para o valor de $log \ p_\theta(X)$ que depende dos conjuntos

de parâmetros ϕ e θ. Com isso, o processo de treinamento é capaz de maximizar a verossimilhança marginal com relação a ϕ e θ, possibilitando o treinamento do codificador e do decodificador de forma conjunta (GIRIN *et al.*, 2020).

Conforme mostrado em Kingma e Welling (2019), é possível reescrever a verossimilhança marginal $log\ p_\theta(x)$ de acordo com a equação:

$$\begin{aligned} log\ p_\theta(x) &= \mathbb{E}_{q_\phi(z|x)}[log\ p_\theta(x)] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[log\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right]\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[log\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right] \\ &= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[log\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]\right]}_{\mathcal{L}_{\theta,\phi}(x)\ \text{(ELBO)}} + \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[log\left[\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right]}_{D_{KL}(q_\phi(z|x)||p_\theta(z|x))}, \end{aligned} \tag{2.19}$$

em que $D_{KL}(q_\phi(z|x)||p_\theta(z|x))$ é a divergência de Kullback-Leibler (KL)[2] entre $q_\phi(z|x)$ e $p_\theta(z|x)$, que é um termo maior ou igual a zero. Sendo assim, o limite inferior de $log\ p_\theta(x)$ é dado pelo primeiro termo da última linha da Equação 2.19, conhecido como limite inferior de evidência - *evidence lower bound* (ELBO) - ou limite inferior variacional.

Ainda, tomando apenas o termo $\mathcal{L}_{\theta,\phi}(x)$, é possível reescrevê-lo da seguinte forma:

$$\begin{aligned} \mathcal{L}_{\theta,\phi}(x) &= \mathbb{E}_{q_\phi(z|x)}\left[log\ p_\theta(x,z)\right] - \mathbb{E}_{q_\phi(z|x)}\left[log\ q_\phi(x,z)\right] \\ &= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[log\ p_\theta(x,z)\right]}_{\text{Acurácia de reconstrução}} - \underbrace{D_{KL}(q_\phi(z|x)||p_\theta(z))}_{\text{Regularização}}. \end{aligned} \tag{2.20}$$

Conforme mencionado por Girin *et al.* (2020), o primeiro termo da Equação 2.20 corresponde à acurácia média do autocodificador, enquanto que o segundo termo atua como um regularizador, forçando que a distribuição a posteriori $q_\phi(z|x)$ se aproxime da distribuição a priori $p_\theta(z)$. Sendo assim, esse termo força que a variável latente se torne independente e, com isso, codifique características diferentes dos dados de entrada.

A maximização da expressão de $\mathcal{L}_{\theta,\phi}(x)$ permite então a otimização dos parâmetros ϕ e θ do VAE. O termo de regularização na Equação 2.20 possui expressão analítica se utilizadas algumas das distribuições usuais, enquanto que a esperança sobre $q_\phi(z|x)$ é intratável, o que leva a estimar este termo com o Método de Monte Carlo (KINGMA; WELLING, 2013; KINGMA; WELLING, 2019; GIRIN *et al.*, 2020). Com isso, tomando N amostras independentemente e identicamente amostradas de $q_\phi(z|x)$, é possível aproximar

[2] Pode ser interpretada como uma distância estatística entre duas distribuições de probabilidade. Para mais informações, se referir a Markatou, Karlis e Ding (2021).

a esperança intratável e com isso calcular $\mathcal{L}_{\theta,\phi}(x)$ para todo o conjunto de dados se z for amostrado como uma função dos parâmetros θ, conforme as equações abaixo:

$$\mathbb{E}_{q_\phi(z|x)}\left[log\ p_\theta(x,z)\right] \approx \frac{1}{N}\sum_{r=1}^{N} log\ p_\theta(x,z^{(r)}), \tag{2.21}$$

$$z = \mu_\phi(x) + \sigma_\phi(x)\epsilon,\ \epsilon \sim \mathcal{N}(0,1). \tag{2.22}$$

Kingma e Welling (2013) pontuam ainda que, durante a otimização pelo método do gradiente descendente estocástico, se o tamanho do lote de amostras for suficientemente grande (maior que 100), é possível tomar $N = 1$ na Equação 2.21. Já a expressão 2.22 é descrita como truque de reparametrização, que permite descrever a variável aleatória z dada pela probabilidade condicional $q_\phi(z|x)$ como uma função determinística $z = g_\phi(x,\epsilon)$, com ϵ uma variável auxiliar e $g_\phi(.)$ uma função vetorial parametrizada por θ. Para mais propriedades matemáticas e derivações a respeito dos VAEs, referir-se aos trabalhos de Kingma e Welling (2013), Rezende, Mohamed e Wierstra (2014), Kingma e Welling (2019), Girin *et al.* (2020).

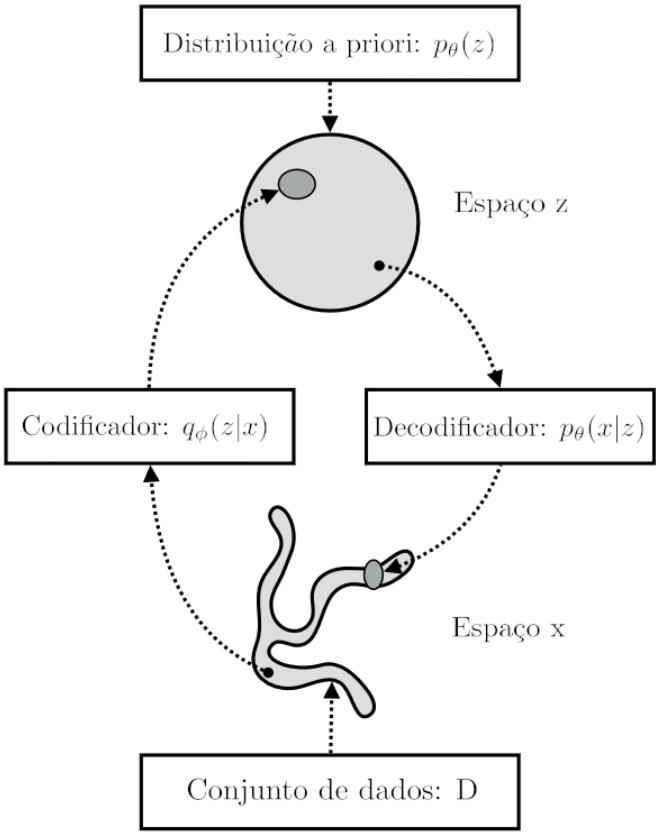
A Figura 8 ilustra resumidamente o funcionamento por trás de um VAE, conforme explicado anteriormente. A partir de um conjunto de dados, de distribuição complicada, o codificador aprende a mapear, de forma estocástica, o espaço referente ao conjunto de dados (espaço x) para um espaço mais simples, definido pelo projetista e chamado de variável latente. O decodificador, treinado de forma conjunta, aprende o mapeamento reverso, sendo capaz de gerar saídas no espaço x a partir de amostras do espaço z. Isso é atingido a partir da otimização do ELBO por meio do método de gradiente descendente estocástico.

2.3 Estimativa de incerteza no aprendizado profundo

O emprego de modelos de aprendizado profundo representa um interesse crescente em todos os tipos de problema de inferência e de tomada de decisão. Essa atenção dada aumenta a necessidade de avaliar a confiabilidade e a eficácia desses modelos antes que eles possam ser postos em prática, já que as predições realizadas estão sujeitas a ruídos e a erros de inferência (ABDAR *et al.*, 2021).

Existem dois tipos principais de incerteza que podem ser modelados no campo do aprendizado profundo: a incerteza aleatória, que captura o ruído inerente às observações (dados), e a incerteza epistêmica, que se refere à incerteza nos parâmetros do próprio modelo (KENDALL; GAL, 2017). Cada tipo pode ser ainda dividido entre a incerteza homocedástica, que se mantém constante para diferentes entradas, e a incerteza heterocedástica, que depende das entradas do modelo. À soma da incerteza aleatória com a incerteza epistêmica é dado ao nome de incerteza preditiva.

Figura 8 – Ilustração do funcionamento de um VAE. A arquitetura aprende um mapeamento estocástico entre um espaço x observado, geralmente de distribuição complicada, e um espaço z, chamado de variável latente, cuja distribuição pode ser simples.



Fonte: Figura adaptada de Kingma e Welling (2019).

A incerteza heterocedástica é especialmente importante para aplicações de visão computacional (KENDALL; GAL, 2017). Quando precisamente quantificada, permite que o sistema de visão computacional lide com os erros do modelo de aprendizado profundo e alcance o melhor desempenho possível (RUSSELL; REALE, 2021).

2.3.1 Incerteza aleatória heterocedástica

A incerteza aleatória heterocedástica modela a FDP $p(Y|X)$ para um dado conjunto θ de pesos de uma rede neural (GAL; GHAHRAMANI, 2016a; RUSSELL; REALE, 2021). A estimativa desse componente de incerteza é feito por meio da maximização da probabilidade a *posteriori* (MAP) de um dado modelo estatístico (GAL; GHAHRAMANI, 2016b).

Para uma rede neural qualquer, é possível interpretar o conjunto de entradas X como um conjunto de vetores independentes de dimensão igual ao número de variáveis de entrada (n). Assumindo que cada um dos vetores de entrada pode ser aproximado por uma

Gaussiana, segue que $X \sim \mathcal{N}(\mu(X), \Sigma(X))$. Dessa forma, $\mu(X)$ é o conjunto de predições convencional do modelo, que também pode ser denotado por $f(X)$ e $\Sigma(X)$ é uma saída adicional que descreve a matriz de covariâncias das variáveis preditas por $f(x)$.

Calculando então a probabilidade de todas as variáveis de entrada (consideradas independentes) se situarem num mesmo valor hipotético, conhecida como a função de probabilidade conjunta ($P_{entrada}$), chega-se em:

$$P_{entrada}(x|\mu(x), \Sigma(x)) = \prod_{i=0}^{n} f(x_i|\mu(x_i), \Sigma(x_i)). \tag{2.23}$$

Ao aplicar o logaritmo natural na expressão e substituir a definição de uma função Gaussiana multivariada, chega-se nas equações:

$$l(x|\mu(x), \Sigma(x)) = log \prod_{i=0}^{n} f(x_i|\mu(x_i), \Sigma(x_i)), \tag{2.24}$$

$$l(x|\mu(x), \Sigma(x)) = log \prod_{i=0}^{n} \frac{1}{(2\pi)^p |\Sigma(x_i)|^{\frac{1}{2}}} exp(-\frac{1}{2}(x_i - \mu(x_i))^T \Sigma^{-1}(x_i - \mu(x_i))), \tag{2.25}$$

$$l(x|\mu(x), \Sigma(x)) = -\frac{1}{2} \sum_{i=0}^{n} [log(2\pi) + log|\Sigma| + (x_i - \mu(x_i))^T \Sigma^{-1}(x_i)(x_i - \mu(x_i))]. \tag{2.26}$$

Para aplicar a MAP, basta maximizar a Equação 2.26.

Entretanto, alguns termos dessa equação são constantes e, durante o treinamento de uma rede neural, é mais comum buscar minimizar uma função de custo. Sendo assim, invertendo o sinal da equação, removendo os termos constantes e reescrevendo a expressão em função da predição de uma rede neural, chega-se na função de custo a ser minimizada para que o modelo aprenda a predizer a incerteza aleatória, descrita pela seguinte equação:

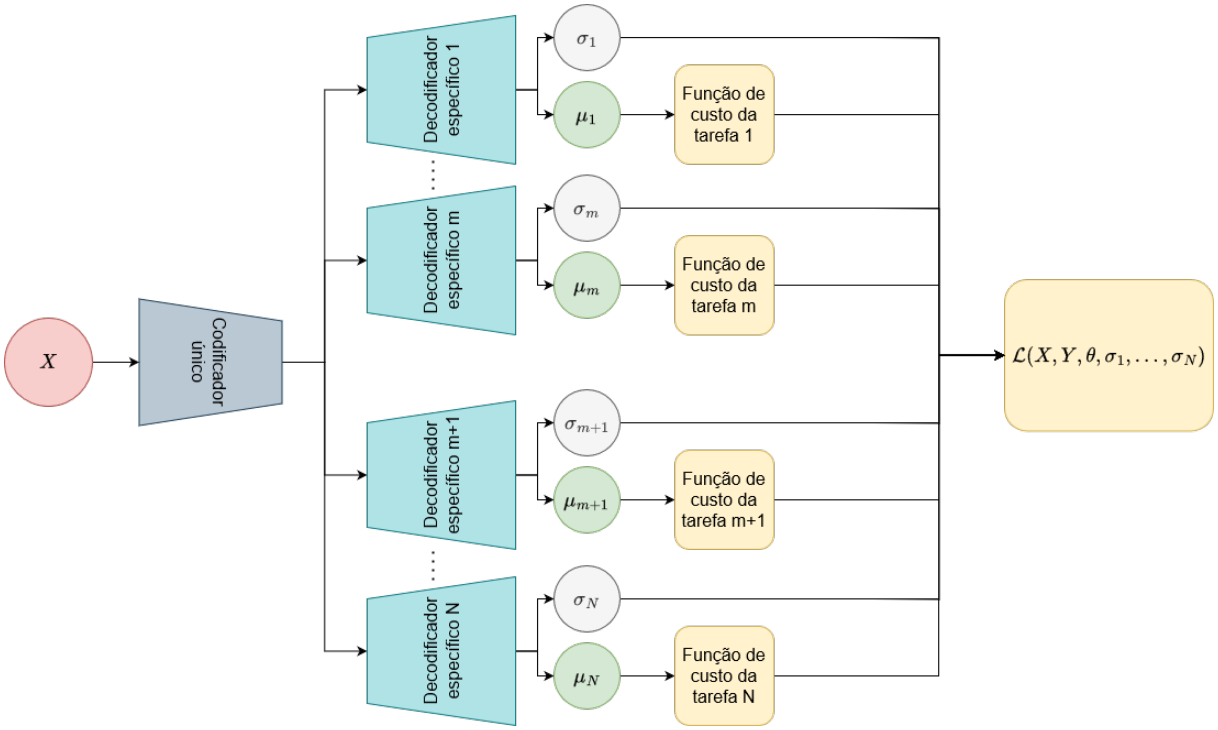
$$\mathcal{L}(x, y) = \frac{1}{2}[log|\Sigma(x)| + (y - \mu(x))^T \Sigma^{-1}(x)(y - \mu(x))]. \tag{2.27}$$

2.3.2 Incerteza aleatória homocedástica

A incerteza aleatória homocedástica captura incertezas relacionadas a informações que o conjunto de dados não consegue explicar, mas que se mantém constante ao longo de todas as amostras de entrada e que varia entre diferentes tarefas. Kendall, Gal e Cipolla (2018) modelaram a incerteza aleatória homocedástica para um modelo de aprendizado profundo capaz de predizer, ao mesmo tempo, a segmentação semântica, a segmentação das instâncias e a predição de profundidade de uma cena.

Além dessas saídas, o modelo predizia, para cada tarefa, a variância da incerteza relacionada a essa tarefa, que é utilizada como forma de ponderar a contribuição da função de custo de cada tarefa na função de custo final do modelo. A Figura 9 ilustra uma rede neural multitarefas, as entradas, as saídas e como as predições, incertezas preditas e funções de custo específicas a cada tarefa se conectam na função de custo total conforme proposto por Kendall, Gal e Cipolla (2018).

Figura 9 – Diagrama esquemático da metodologia proposta por Kendall, Gal e Cipolla (2018) para modelar e utilizar a incerteza aleatória homocedástica como forma de balancear funções de custo de m tarefas de classificação e $N - m$ tarefas de regressão para uma rede neural profunda multitarefas.



Fonte: Elaborada pelo autor.

A modelagem proposta é similar àquela feita para modelar a incerteza aleatória heterocedástica, na medida em que a MAP também é maximizada, mas considerando múltiplas saídas diferentes e independentes. Exemplificando, para uma rede neural capaz de predizer duas tarefas univariadas de regressão, a seguinte equação mostra a função de custo a ser minimizada considerando y_i e $f_i(.)$ o conjunto de rótulos e a predição do modelo para $i-$ésima tarefa:

$$\mathcal{L}(x, y) = \frac{1}{2\sigma_1^2}(y_1 - f_1(x))^2 + \frac{1}{2\sigma_2^2}(y_2 - f_2(x))^2 + log\sigma_1\sigma_2. \tag{2.28}$$

Já para uma tarefa de regressão hipotética 1 e outra tarefa hipotética 2 de classificação, a função de custo a ser minimizada é dada por:

$$\mathcal{L}(x,y) = \frac{1}{2\sigma_1^2}(y_1 - f_1(x))^2 - \frac{1}{\sigma_2^2} log(Softmax(y_2, f_2(x))) + log\sigma_1 + log\sigma_2. \quad (2.29)$$

2.3.3 Incerteza epistêmica heteroscedástica

A incerteza epistêmica heteroscedástica aproxima a FDP $P(W|X,Y)$, ou seja, a probabilidade dos parâmetros θ estimados durante o treinamento de um modelo dado o conjunto de evidências (dados) utilizados durante o treinamento, ou seja, os conjuntos X e Y. Dado que durante o treinamento de uma rede neural de aprendizado profundo é altamente provável que o conjunto de pesos obtidos seja fruto de um novo mínimo local a cada novo treinamento com valores iniciais distintos, este componente de incerteza pode ser modelado avaliando um mesmo modelo para diferentes conjuntos θ.

Dentre as técnicas conhecidas para gerar múltiplos conjuntos θ, destacam-se o Monte Carlo *Dropout* (GAL; GHAHRAMANI, 2016a) e a criação de conjuntos de modelos de aprendizado profundo (*deep ensembles*) (ABDAR *et al.*, 2021) para uma única tarefa. Uma vez escolhido o método para geração de múltiplas amostras de θ, basta aplicar a seguinte equação durante a avaliação do modelo para obter o componente de incerteza em questão:

$$\Sigma^{epi} \approx \frac{1}{N}\sum_{n=1}^{N} f_n(x) f_n(x)^T - \frac{1}{N^2}(\sum_{n=1}^{N} f_n(x))(\sum_{n=1}^{N} f_n(x))^T. \quad (2.30)$$

Em relação ao número de amostras necessárias para avaliar o componente de incerteza, Gal e Ghahramani (2016b) sugerem $N = 50$ amostras.

2.3.3.1 *Deep Ensembles*

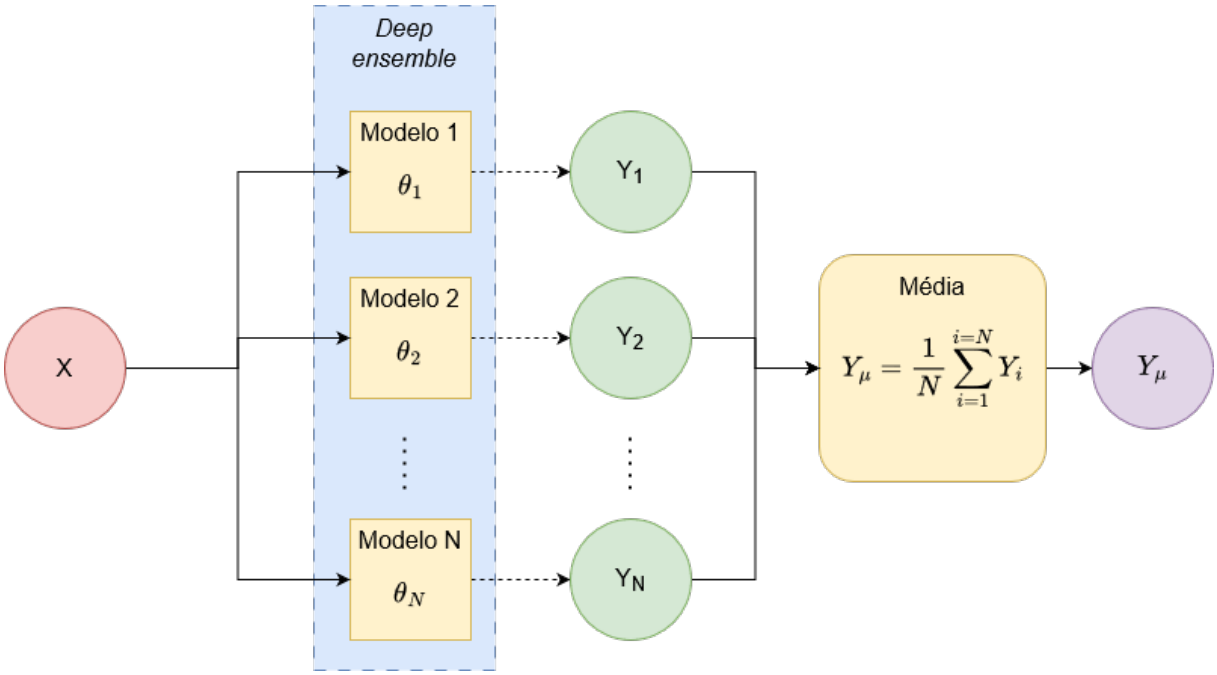
A Figura 10 ilustra como é criado um *deep ensemble* para avaliação da incerteza epistêmica heterocedástica de uma rede neural. Afim de obter amostras diferentes do modelo em relação a possíveis conjuntos de pesos, N diferentes treinamentos são feitos para uma dada rede neural, formando um conjunto de N modelos para uma dada arquitetura.

Uma vez obtidos os modelos, durante a avaliação de um conjunto de entradas (X), o conjunto de saída (Y) é calculado a partir da média das saídas (y_i) de cada um dos modelos. As saídas y_i são utilizadas na Equação 2.30 para calcular Σ^{epi}.

2.3.3.2 Monte Carlo *Dropout*

Esta técnica de amostragem foi proposta por Gal e Ghahramani (2016a) com base na técnica de *dropout*, geralmente utilizada para regularizar redes neurais durante o treinamento, evitando o sobreajuste do modelo aos dados (GOODFELLOW; BENGIO; COURVILLE, 2016). A Figura 11 ilustra o funcionamento da técnica de *dropout*, que

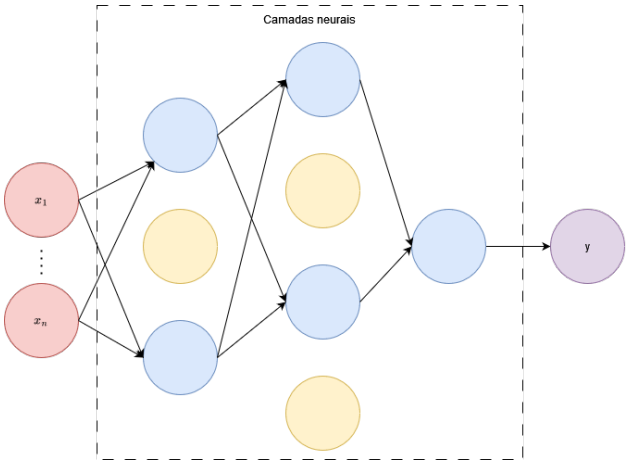
Figura 10 – Diagrama esquemático correspondente a um *deep ensemble* composto por N modelos treinados de uma rede neural, cada um com um conjunto de pesos θ.



Fonte: Elaborada pelo autor.

corresponde a definir uma probabilidade de excluir os pesos de cada neurônio em camada anterior à de saída durante o treinamento.

Figura 11 – Diagrama esquemático de uma rede neural artificial com *dropout* ativado para a primeira e segunda camadas neurais. Neurônios em azul estão ativados, enquanto neurônios em amarelo estão desativados.



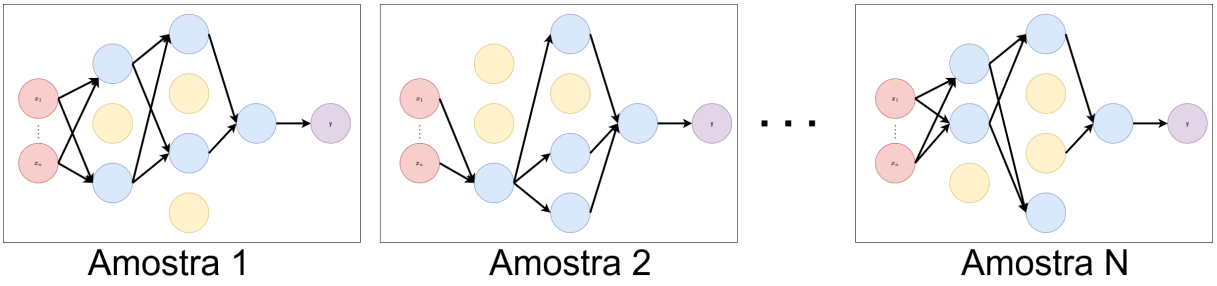
Fonte: Elaborada pelo autor.

Ao amostrar quais neurônios estão ativos a cada passo de treinamento e considerando que o conjunto de treinamento é suficientemente grande, o *dropout* pode ser pensado como uma técnica que permite formar um grupo de todas as sub-redes que podem ser formadas a

partir de uma arquitetura de rede neural (GOODFELLOW; BENGIO; COURVILLE, 2016). Antes só utilizado durante o treinamento de uma rede neural, o *dropout* foi aplicado por Gal e Ghahramani (2016a) também durante a inferência do modelo, criando dessa forma, um conjunto de múltiplas sub-redes de um modelo. Dessa forma, os autores mostraram que o método se aproxima de uma inferência Bayesiana em processos Gaussianos profundos, permitindo estimar a incerteza epistêmica heterocedástica do modelo.

A Figura 12 mostra uma representação do método de amostragem Monte Carlo *Dropout*, proposto por Gal e Ghahramani (2016a). Ao contrário do método ilustrado na Figura 10 em que N diferentes modelos são utilizados para gerar o conjunto de saídas, na Figura 12, um único conjunto de pesos é necessário, ou seja, basta que apenas um treinamento seja realizado. Novamente, o conjunto de saída é obtido pela média das saídas de cada amostra, dada por uma sub-rede.

Figura 12 – Ilustração da técnica de amostragem Monte Carlo *Dropout*, na qual N amostras de predições são obtidas a partir da técnica de *dropout* durante a predição de um único conjunto de pesos θ da arquitetura.



Fonte: Elaborada pelo autor.

2.4 Considerações finais

Dados os conceitos apresentados neste Capítulo, este trabalho se propõe a desenvolver um modelo de rede neural de aprendizado profundo para predição de sequência, que, neste caso, consiste em trajetórias de pedestres. Sendo assim, as arquiteturas mais relevantes para a compreensão do trabalho são a das redes LSTM e Transformadoras.

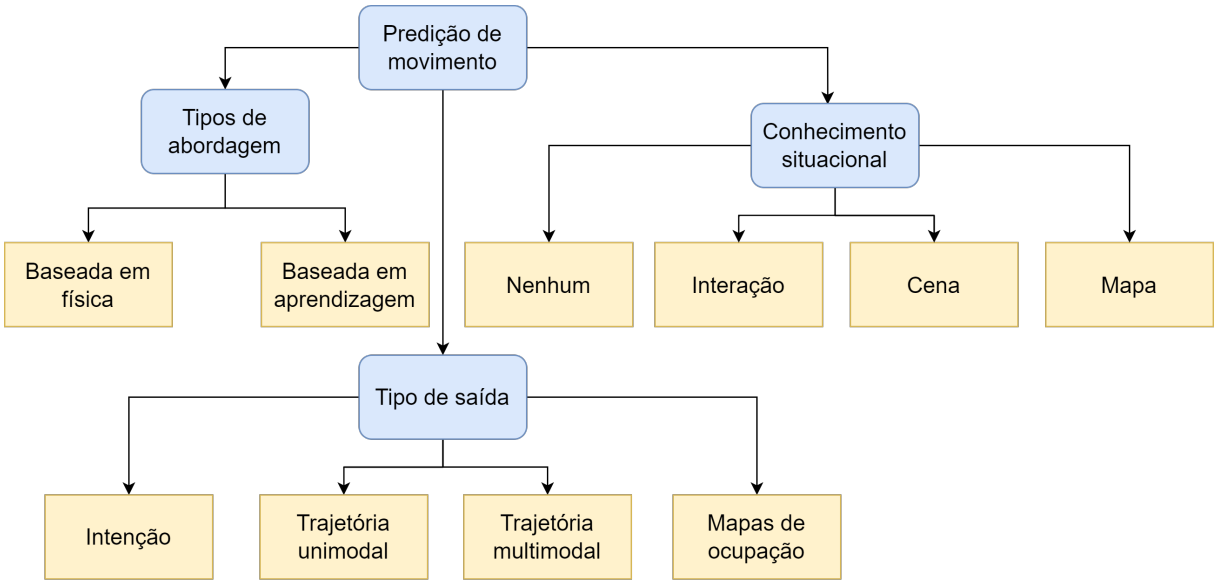
Além disso, do ponto de vista da estimativa de incerteza no aprendizado profundo, este trabalho modela a incerteza preditiva heterocedástica do modelo proposto, composto pela soma das incertezas aleatórias e epistêmicas heterocedásticas, conforme apresentado anteriormente. A abordagem possui ainda uma diferença conceitual em relação a aplicação de VAEs, que são arquiteturas cujas saídas também modelam funções densidade de probabilidade. Essa diferença e as contribuições propostas são discutidas nos próximos Capítulos, após a revisão bibliográfica dos trabalhos na área de aplicação.

3 PREDIÇÃO DE MOVIMENTO DE PEDESTRES

Este capítulo apresenta uma revisão dos trabalhos relacionados à temática de predição de movimento de pedestres utilizando aprendizado profundo. Para isso, este trabalho se baseia na taxonomia apresentada por Gulzar, Muhammad e Muhammad (2021) para classificar os métodos de predição de movimento dos agentes do trânsito.

A Figura 13 mostra um diagrama esquematizando como os métodos de predição de movimento podem ser classificados. Dessa forma, este trabalho se refere a métodos de predição de trajetória cuja abordagem é baseada em aprendizagem (redes neurais artificias) e não em modelagem (como filtros de Kalman). Além disso, em relação ao conhecimento do ambiente possuído pelos métodos, por este se tratar de um trabalho no escopo da predição a partir de câmeras móveis, os métodos revisados mais relevantes para o trabalho apresentam apenas conhecimentos da cena e de interação entre os agentes.

Figura 13 – Diagrama esquemático da taxonomia proposta por Gulzar, Muhammad e Muhammad (2021) para a predição de movimento de agentes do trânsito.



Fonte: Figura adaptada de Gulzar, Muhammad e Muhammad (2021).

Por fim, com relação ao tipo de saída, este problema é abordado de duas diferentes formas na literatura: predição de intenção e predição de trajetória (na forma unimodal, multimodal ou de mapa de ocupação). No primeiro caso, o objetivo é antecipar a ação a ser tomada por cada pedestre na cena a partir da classificação da ação do pedestre (andando, atravessando a rua ou parado, por exemplo). Já na abordagem de predição de trajetória, como o próprio nome já diz, o intuito é calcular possíveis posições futuras do pedestre a partir do quadro atual e de uma janela de quadros passados. Tanto a janela de observação

quanto a duração do horizonte de predição costumam variar de acordo com o conjunto de dados abordado, mas é razoável considerar que, para a implementação num veículo autônomo, o horizonte de tempo das estimativas futuras seja maior do que o tempo de reação desses sistemas.

3.1 Predição de trajetória

Em relação ao desenvolvimento de veículos antônomos, que tem crescido nos últimos anos, ambientes densamente urbanizados continuam a oferecer um grande desafio à implementação dessa tecnologia (BHATTACHARYYA *et al.*, 2021). Nesses cenários, a capacidade de antecipação é um dos componentes chaves para o sucesso dos veículos autônomos, sendo a capacidade de antecipação do movimento de participantes do trânsito, como pedestres e outros veículos, desafiadora devido à interação entre esses agentes (BHATTACHARYYA *et al.*, 2021).

No caso dos pedestres, que representam a classe mais suscetível a acidentes dentre as envolvidas no trânsito, o desenvolvimento de sistemas de proteção que reduzam ou mitiguem os impactos a esses indivíduos se torna um tema importante para veículos autônomos (SIGHENCEA; STANCIU; CĂLEANU, 2021). Nesse contexto, em comparação à predição de intenção, a predição de trajetória possibilita incorporar o conhecimento de como o ambiente irá mudar no futuro próximo ao veículo de uma maneira pró-ativa, com o aprimoramento das tarefas de percepção ativa, planejamento preditivo de trajetória, controle preditivo e de interação humano-robô (RUDENKO *et al.*, 2020).

A predição de trajetória, que tem sido foco de muitas pesquisas nos últimos anos, continua sendo um problema desafiador, visto que as predições podem depender tanto de fatores ambientais quanto internos ao indivíduo, como a intenção e os objetivos (RUDENKO *et al.*, 2020; WANG *et al.*, 2022). Neste sentido, Sighencea, Stanciu e Căleanu (2021) apresentam uma revisão bibliográfica recente sobre o tema de aprendizado profundo para a predição de trajetória de pedestres, abordando os tipos de sensores utilizados em relação às técnicas de aprendizado profundo, uma análise dos principais métodos desse campo de pesquisa, as limitações da literatura e os possíveis novos temas de pesquisa, além de abordar os conjuntos de dados disponíveis.

Neste ponto, é importante ressaltar que a predição de trajetória futura de pedestres não se limita apenas ao escopo de veículos autônomos, existindo ainda aplicações como o desenvolvimento de robôs de serviços, vigilância de tráfego e monitoramento de pedestres em ambientes movimentados (BECKER *et al.*, 2018; RUDENKO *et al.*, 2020; SIGHENCEA; STANCIU; CĂLEANU, 2021). Dessa forma, existem conjuntos de dados disponíveis que divergem com relação à forma em que as imagens de entrada (e consequentemente as trajetórias) são calculadas, indo de imagens aéreas, imagens obtidas por câmeras fixas e imagens obtidas por câmeras acopladas a carros, além de divergências entre coordenadas

no plano da imagem e coordenadas no mundo.

Apesar deste trabalho dar preferência à definição do problema a partir de imagens obtidas por câmeras acopladas a veículos, métodos desenvolvidos e avaliados em outros conjuntos de dados podem fornecer técnicas e resultados satisfatórios para aplicações automotivas (SIGHENCEA; STANCIU; CĂLEANU, 2021). Além disso, em um cenário de implantação total de veículos autônomos e de integração com cidades inteligentes, a possibilidade de se construir um mapa, de comunicação com a infraestrutura da cidade ou de cooperação com robôs aéreos permitiria tratar o problema sem levar em conta o deslocamento da câmera juntamente do movimento do pedestre (RIDEL *et al.*, 2020).

O primeiro trabalho a abordar o problema de predição de trajetória de pedestres a partir de uma câmera móvel foi apresentado por Bhattacharyya, Fritz e Schiele (2018). Os autores discutem as diferenças desse problema em relação ao cenário em que a câmera é fixa e argumentam que o movimento do veículo em que a câmera está acoplada é mais dominante sobre as trajetórias futuras que aspectos de interação social entre os pedestres da cena.

Sendo assim, este trabalho se assemelha às revisões bibliográficas feitas por Chen e Tian (2021), no que diz respeito à predição de trajetória, e por Sighencea, Stanciu e Căleanu (2021), quando aborda métodos baseados em câmeras acopladas a veículos. Ainda, o trabalho de Ridel *et al.* (2018) apresenta uma revisão bibliográfica da área de predição de comportamento de pedestres, destacando conjuntos de dados disponíveis e uma tabela com entradas, saídas e os métodos utilizados pela literatura até o momento da publicação.

3.2 Definição do problema

Seguindo a notação de Chen e Tian (2021), o problema pode ser definido como encontrar as trajetórias futuras $Y_{pred} = y_{t+1}, ..., y_{t+T_h}$, com T_h o horizonte de predição, para cada agente na cena dado o conjunto de observações $X_{obs} = \{x_{t-T_o}, ..., x_t\}$ composto pelas observações passadas para o agente dentro da janela de observação temporal de duração T_o. Por se tratar de uma formulação em que um único conjunto Y_{pred} é obtido a partir da sequência se observações, essa abordagem diz respeito a modelos cujo tipo de saída é uma trajetória unimodal, conforme a taxonomia esquematizada na Figura 13.

Já para abordagens em que o tipo de saída é uma FDP $p(.)$ representando a distribuição futura da posição do agente para o instante de tempo $t + 1$ até $t + T_h$, o objetivo é encontrar $p_\theta(Y_{pred}|X_{obs})$, com θ o conjunto de parâmetros do modelo. Essa abordagem probabilística do problema serviu tanto para Bhattacharyya, Fritz e Schiele (2018) predizerem trajetórias futuras modelando a incerteza da predição usando técnicas de aprendizado Bayesiano quanto para trabalhos como Salzmann *et al.* (2020), Yao *et al.* (2021), e Bhattacharyya *et al.* (2021) predizerem trajetórias multimodais a partir de

múltiplas amostras da FDP de saída.

3.2.1 Possíveis conjuntos de observações

Imagens capturadas por câmeras, devido à riqueza de informações e à relativa facilidade de se obter esse tipo de dados em relação a outros métodos, são um dos componentes do conjunto de observação mais escolhidos para compor o conjunto de observações. Ainda assim, outras informações podem ser utilizadas na tarefa, mesmo que sejam calculadas a partir da imagem de entrada. Sendo assim, esta seção discute quais são as entradas geralmente utilizadas na literatura e a justificativa para cada uma delas.

Começando pela trajetória passada do agente, Chen e Tian (2021) apresentam uma breve análise dos dois métodos mais comumente utilizados nesse caso: (i) representar cada uma das posições passadas por um único ponto referente à cabeça, ao tronco ou ao centro entre os dois pés do pedestre, ou (ii) representar as posições pelas coordenadas de um retângulo (ou cubo no caso de uma representação espacial) contendo o objeto, composto tanto pelas coordenadas dos vértices dessa figura ou na forma de alguns vértices e medidas de distâncias. Enquanto a primeira forma de representação, que também pode ser calculada a partir da segunda, pode ser mais precisa, o emprego da representação por forma geométrica oferece mais informações contextuais para o modelo (como mudanças no tamanho relativo dos agentes) e consegue representar melhor situações de oclusão parcial.

Métodos baseados em câmeras acopladas a veículos capturam uma perspectiva frontal da cena, a partir da qual é possível obter detalhes mais claros dos pedestres detectados. Além disso, como para esses conjuntos as posições detectadas estão descritas no plano da imagem, é comum a inclusão de entradas que representam, de alguma forma, o movimento do veículo que carrega a câmera.

Dessa forma, o trabalho de Rasouli *et al.* (2019) adicionou como entrada do conjunto de observação uma imagem de contexto, definida como uma região da imagem de entrada de tamanho fixo centrada na posição do pedestre detectado, de modo que informações visuais do ambiente, como placas, semáforos e faixas de pedestres próximas ao agente em questão possam ser codificadas pelo modelo. Além disso, Rasouli *et al.* (2019) adicionaram ao modelo de predição de trajetória um módulo que prediz a intenção do pedestre de atravessar ou não a via com base também na trajetória passada e no contexto local.

Já os trabalhos de Cao e Fu (2020), de Santos e Grassi (2021), de Su *et al.* (2022) e de Czech *et al.* (2023) utilizaram como entrada a pose do pedestre descrita. Nos dois primeiros, a informação foi obtida pelo OpenPose, um método eficiente para extração de pose bidimensional humana em tempo real e de forma invariante ao número de pessoas detectadas na cena (Cao *et al.*, 2019). Em métodos como esse, os pontos que descrevem a pose geralmente são distribuídos ao longo do corpo humano formando um esqueleto. A Figura 14 mostra um exemplo da detecção do OpenPose numa cena com diversos humanos

em que é possível ver com o a distribuição desses pontos ao longo do corpo consegue refletir a ação sendo feita para aquele instante de tempo.

Figura 14 – Exemplo de detecção de pose humana bidimensional utilizando o OpenPose (Cao *et al.*, 2019).

Fonte: Figura extraída de Cao *et al.* (2019).

Além da informação de pose dos pedestres, Cao e Fu (2020), Santos e Grassi (2021) e Su *et al.* (2022) utilizam também a velocidade escalar do veículo de referência (que carrega a câmera) obtida a partir dos sensores do carro. Por não ser uma descrição vetorial, essa entrada acaba atuando mais como uma descrição da interação entre o pedestre e o veículo (o pedestre tende a atravessar a rua se o veículo parar, por exemplo) do que como uma descrição do movimento da câmera.

De forma semelhante, Su *et al.* (2022) adicionaram também a orientação do veículo e uma anotação do pedestre estar olhando para o veículo com a câmera no último instante observável, afim de melhorar o conjunto de informações que sugerem interação entre os agentes. Já Czech *et al.* (2023) estimaram a trajetória do veículo com base nas anotações de velocidade escalar e de orientação usando um modelo de velocidade constante.

Já os trabalhos de Bhattacharyya, Fritz e Schiele (2018), Ridel *et al.* (2019) e de Cai *et al.* (2021), utilizaram conjuntos de dados que possibilitaram ao modelo considerar o movimento da câmera. Bhattacharyya, Fritz e Schiele (2018) utilizaram como conjunto de observação, além da trajetória passada dos pedestres, dados de odometria do veículo de referência e a imagem RGB de entrada, a partir da qual era extraída também a odometria visual. Ridel *et al.* (2019) utilizaram a trajetória passada do pedestre no espaço, a orientação da cabeça dos pedestres como uma forma de aprender de forma implícita a interação entre o pedestre e o veículo, e a trajetória do veículo. Todos os dados do conjunto de entradas foram obtidos por meio de anotações feitas no conjunto de dados abordado, que foi coletado a partir de vídeos baseados em sistemas de visão estéreo.

Cai *et al.* (2021) utilizaram um conjunto de dados próprio para o qual foram coletadas as informações de posição e de velocidade da câmera juntamente com as imagens RGB. Dessa forma, os autores realizam a predição da trajetória dos pedestres a partir desses dados mencionados.

Já os métodos de predição de trajetória de pedestres a partir de imagens obtidas por câmeras fixas são maioria na literatura, não só pelo fato desses conjuntos de dados estarem disponíveis a mais tempo, mas também dado que imagens aéreas obtidas por câmeras de seguranças ou por *drones* podem ser utilizadas, por exemplo, para vigilância. A essa vista aérea das imagens é dada o nome de perspectiva *bird's eye view* (BEV).

Imagens capturadas na perspectiva BEV permitem a utilização de informações da cena como a segmentação semântica da cena. Nesse sentido, Ridel *et al.* (2020) utilizaram uma rede de segmentação semântica para classificar a superfície da imagem aérea dentre sete possíveis classes: construções, vegetação rasteira, árvore, carro, calçada, superfície impenetrável ou imagem de fundo. Dessa forma, a saída dessa operação agrega uma noção de navegabilidade do pedestre no ambiente. Mangalam *et al.* (2021) utilizaram uma abordagem parecida, classificando cada *pixel* da imagem de entrada em pavimento, estrutura, rodovia, árvore ou terreno não construído.

Salzmann *et al.* (2020) criaram uma representação dos agentes da cena (nós ou vértices) observada e de suas interações (arestas) na forma de um grafo espaço-temporal direcionado. Os nós podem ser qualquer participante do trânsito presente na imagem, como pedestres, carros e ônibus. No grafo, os vértices são ainda rotulados de acordo com a classe que esse agente faz parte. Já as interações são baseadas na distância de cada agente de referência aos demais agentes da cena. Caso esse valor seja menor do que um limiar definido para cada classe, uma aresta direcionada é adicionada apontando para o agente de referência, indicando que o outro interfere em seu movimento. Dessa forma, o conjunto de observação construído pelos autores é composto da sequência temporal passada desses grafos.

Por fim, os trabalhos de Mangalam *et al.* (2020), Giuliari *et al.* (2021), Yao *et al.* (2021), Wang *et al.* (2022) se propõem a utilizar apenas as trajetórias passadas como conjunto de observação. O interessante dessa abordagem é que, no caso dos métodos de Yao *et al.* (2021) e de Wang *et al.* (2022), foi possível que os autores avaliassem os resultados obtidos tanto para conjuntos de dados na perspetiva BEV quando para imagens obtidas a partir da perspectiva do veículo, caracterizando então métodos invariantes à perspectiva da imagem.

A Tabela 1 detalha, para cada um dos métodos citados, as entradas do modelo ou do módulo de predição de trajetória, bem como o tipo de arquitetura de rede neural utilizada e o conjunto de dados abordado.

Tabela 1 – Descrição das entradas, arquitetura de rede neural utilizada e o conjunto de dados de trabalhos de predição de trajetória de pedestres.

Estudo	**Entradas**	**Arquitetura**	**Conjunto de dados**
Bhattacharyya, Fritz e Schiele (2018)	Trajetória, imagens RGB, odometria	LSTM Bayesiana	Cityscapes
Rasouli *et al.* (2019)	Trajetória, contexto, intenção e velocidade escalar do veículo	CNN, ConvLSTM	PIE
Ridel *et al.* (2019)	Trajetória do pedestre e do veículo, orientação da cabeça	LSTM	Fornecido por outros autores
Cao e Fu (2020)	Trajetória, pose, velocidade escalar do veículo e contexto	LSTM, convLSTM, GCN	PIE e conjunto próprio
Cai *et al.* (2021)	Imagens RGB, posição e velocidade da câmera	CNN	Próprio
Santos e Grassi (2021)	Trajetória, velocidade escalar do veículo, pose	LSTM	PIE
Bhattacharyya *et al.* (2021)	Trajetória dos agentes na cena	CVAE	EuroPVI
Ridel *et al.* (2020)	Grade semântica da imagem RGB e grade da trajetória	CNN, convLSTM	Stanford *drone*
Salzmann *et al.* (2020)	Grafo espaço-temporal direcionado dos agentes da cena	CVAE, LSTM, GRU, CNN	NuScenes, ETH+UCY
Mangalam *et al.* (2020)	Trajetória	CVAE	ETH+UCY, Stanford *drone*
Giuliari *et al.* (2021)	Trajetória	Transformadora	TrajNet *challenge*, ETH+UCY
Mangalam *et al.* (2021)	Grade semântica da imagem RGB e grade da trajetória	CNN	ETH+UCY, Stanford *drone*, *Intersection drone dataset*
Yao *et al.* (2021)	Trajetória	GRU, CVAE	JAAD, PIE, ETH+UCY
Wang *et al.* (2022)	Trajetória	GRU, CVAE	JAAD, PIE, ETH+UCY
Su *et al.* (2022)	Trajetória, imagem e orientação do pedestre, olhar para o veículo, velocidade e orientação do veículo	Transformer, CVAE	JAAD, PIE
Czech *et al.* (2023)	Trajetória, pose, orientação do pedestre, trajetória do veículo	GRU	PIE, ECP-*Intention*

Fonte: Elaborado pelo autor.

3.2.2 Classes de métodos de solução

Do ponto de vista de aprendizagem de máquina, os métodos existentes na literatura relacionados ao uso de aprendizado profundo para predição de trajetória de pedestres podem ser divididos entre algoritmos de classificação e algoritmos de regressão. No primeiro conjunto, dentro do qual se inclui os trabalhos de Ridel *et al.* (2020) e de Wang e Papanikolopoulos (2020), a posição do pedestre em cada instante de tempo é codificada na forma de grade binária, em que todos os valores são nulos exceto para a célula ocupada pelo pedestre. Dessa forma, a trajetória é codificada na forma de um tensor tridimensional obtido a partir da concatenação das grades bidimensionais para cada instante de tempo.

Essa abordagem permite modelar o problema como um problema de classificação, uma vez que, busca-se obter na saída um formado de grade semelhante ao da entrada. Para isso, o modelo aprende a classificar se cada uma das células da grade de saída representa ou não a posição do pedestre, por meio da aplicação de uma função de custo como a entropia binária cruzada durante o treinamento.

As grades de saída aprendidas pelo modelo de classificação formam uma espécie de mapa de calor indicando as regiões em que o pedestre deve estar localizado a cada instante de tempo. Dessa forma, cria-se uma representação intuitiva a partir da qual é possível amostrar múltiplas possíveis trajetórias e criar uma saída multimodal, apesar de essas grades de saída não possuírem valor matemático de incerteza.

Já o conjunto de algoritmos de regressão busca aprender as relações entre as variáveis de entrada e a saída, observando-a como uma função matemática contínua. Dessa forma, modelos como os dos trabalhos de Rasouli *et al.* (2019), Yao *et al.* (2021) e Wang *et al.* (2022) têm como saída as coordenadas do pedestre a cada instante de tempo. Para isso, são empregadas funções de custo como o erro quadrático médio entre os valores preditos e reais, a fim de aproximar uma função da outra.

Em relação a essas duas possibilidades, o trabalho de Giuliari *et al.* (2021), que propõe um modelo de predição de trajetória de pedestres, testa as duas abordagens para o mesmo modelo. De acordo com Giuliari *et al.* (2021), tanto a literatura quanto os resultados obtidos pelo seu modelo apontam que tratar o modelo para a tarefa de predição de trajetória de pedestres como um algoritmo de regressão gera resultados mais precisos e satisfatórios que a abordagem de classificação. Entretanto, a abordagem de classificação é uma das formas de se obter saídas multimodais, principalmente em arquiteturas que não possuem módulos específicos para isso.

3.3 Arquiteturas de redes neurais de aprendizado profundo utilizadas

Este trabalho foca em apresentar as arquiteturas de alguns dos métodos citados na Tabela 1, especialmente aqueles que mais contribuíram para o estado da arte no tema

abordado. Entretanto, nessa área, métodos aplicados a problemas de vigilância por vídeo e de predição de trajetória de veículos podem ter mecanismos a serem reutilizados.

Conforme mencionado anteriormente, o trabalho de Bhattacharyya, Fritz e Schiele (2018) foi o primeiro a abordar a predição de trajetória de pedestres a partir de uma câmera acoplada a um veículo móvel. Para isso, os autores se propuseram a modelar tanto a incerteza aleatória (da observação) quanto a incerteza epistêmica (do modelo) com o intuito de fornecer uma medida de incerteza geral da predição.

A LSTM Bayesiana, é composta por um módulo bayesiano de predição de trajetória e por um módulo de odometria. Detalhes a respeito da derivação matemática das incertezas e de como incorporá-las no modelo das redes LSTM foram omitidas deste texto, mas podem ser encontradas no trabalho de Bhattacharyya, Fritz e Schiele (2018).

Para avaliar os resultados obtidos, Bhattacharyya, Fritz e Schiele (2018) propuseram avaliar a predição de trajetória de pedestres utilizando o conjunto de dados Cityscapes (CORDTS *et al.*, 2016), empregado na tarefa de segmentação semântica aplicada a veículos autônomos. Esse conjunto de dados possui anotações de pedestres presentes na cena e na época não havia nenhum conjunto de dados de predição de trajetória de pedestre do ponto de vista de um veículo.

Dessa forma, Bhattacharyya, Fritz e Schiele (2018) estabeleceram comparações do seu método com um filtro de Kalman (método clássico e sem o uso de aprendizado profundo) e com método da época referente ao estado da arte de predição de trajetória de pedestres, mas desenvolvido para predições realizadas a partir de imagens obtidas por uma câmera estática. O trabalho de Bhattacharyya, Fritz e Schiele (2018) não só superou os demais como também mostrou com isso a importância de se estudar o problema de predição de trajetória a partir do ponto de vista do veículo, modelando o movimento da câmera juntamente com o movimento do pedestre.

Mais tarde, Rasouli *et al.* (2019) propuseram um conjunto de dados para predição de trajetória e de intenção de pedestres, juntamente com um modelo de predição de trajetória que também realiza a tarefa de predição de intenção de atravessar a via como forma de auxiliar no treinamento do modelo final. O trabalho de Rasouli *et al.* (2019) se destaca na literatura por introduzir um conjunto de dados novo, com imagens obtidas a partir da perspectiva de um veículo, que possibilita tanto a predição de trajetória quanto a predição de intenção de pedestres. Além disso, o estudo mostra que realizar as duas tarefas de forma conjunta pode ser benéfico ao modelo, juntamente com a adição da informação de velocidade escalar.

Por ter sido o primeiro trabalho para esse conjunto, Rasouli *et al.* (2019) compararam os resultados obtidos com experimentos feitos utilizando o modelo da rede LSTM Bayesiana (BHATTACHARYYA; FRITZ; SCHIELE, 2018) no mesmo conjunto de dados,

que era o estado da arte para predição de trajetória de pedestres a partir da perspectiva de um veículo.

Após a proposição do PIE_{traj}, Cao e Fu (2020) e Santos e Grassi (2021) propuseram modelos de predição de trajetória e compararam os resultados obtidos com os de Rasouli *et al.* (2019) e Bhattacharyya, Fritz e Schiele (2018) no conjunto de dados PIE. Cao e Fu (2020) substituíram apenas o módulo de intenção proposto por Rasouli *et al.* (2019) por um outro baseado na pose do pedestre e na relação entre os grupos de pontos que a compõem, enquanto Santos e Grassi (2021) propuseram uma arquitetura um pouco diferente, mas baseada em um módulo de predição de trajetória influenciado pela pose do pedestre e pela velocidade escalar do veículo.

Como contribuição, o trabalho de Santos e Grassi (2021) apresenta um modelo que é mais leve que o proposto por Rasouli *et al.* (2019) e ainda propõe um cenário de avaliação de modelos baseado apenas em dados não rotulados.

Ridel *et al.* (2020) propuseram um modelo para predição de trajetórias multimodais para imagens obtidas por câmeras fixas e na perspectiva BEV. Conforme já mencionado, o modelo se baseia em grades da entrada e das trajetórias passadas. Com isso, os autores empregaram a segmentação semântica da cena a partir de um modelo treinado em outro conjunto de dados para classificar a superfície visível.

Pelo formato de entrada escolhido, Ridel *et al.* (2020) abordam o tema como um problema de classificação, possibilitando, portanto, a geração de trajetórias multimodais mesmo com o método sendo baseado em CNNs e LSTMs. Nessa abordagem, a média da distância média entre as coordenadas preditas e reais a cada quadro futuro é calculada para cada uma das trajetórias multimodais preditas e o melhor valor (o mais baixo) é utilizado como função de perda durante o treinamento.

Como resultado, Ridel *et al.* (2020) conseguiram superar as métricas reportadas pelo estado da arte no momento da publicação do trabalho e a análise qualitativa dos resultados obtidos provou que as trajetórias preditas pelo modelo são coerentes com a trajetória observada e com a cena. Entretanto, os autores indicam como limitação do método o grande custo computacional envolvido, o que pode inviabilizar a aplicação do modelo.

Salzmann *et al.* (2020) propuseram um modelo conhecido como *Trajectron++*, que tem como característica predizer a trajetória de múltiplos e diferentes tipos de agentes da cena, levando em consideração as características dinâmicas de cada um e informações do ambiente. Sendo assim, diferentemente dos outros métodos revisados que se concentram unicamente no movimento de pedestres, o *Trajectron++* prediz trajetórias de todos os agentes da cena, incluindo veículos. Apesar de, por essa característica, o modelo fugir um pouco do escopo deste trabalho, ele é comumente citado por outros métodos de predição

de trajetória de pedestres e por isso foi incluído nesta revisão.

Conforme mencionado na seção 3.2.1, o método de Salzmann *et al.* (2020) tem como entrada um grafo dos agentes na cena. Diferentemente do trabalho de Ridel *et al.* (2020) que aborda a questão da multimodalidade considerando a predição de trajetória como um problema de classificação, Salzmann *et al.* (2020), baseados na literatura, mantém a abordagem de regressão e baseiam seu algoritmo no uso de um CVAE (*conditional variational autoencoder*), que modelam a multimodalidade de forma explícita, ao invés das GANs (*generative adversarial networks*), que modelam a multimodalidade de forma implícita (SALZMANN *et al.*, 2020; YAO *et al.*, 2021).

Juntamente com o grafo de entrada, o *Trajectron++* prevê codificar ainda informações semânticas locais da cena, como mapas de alta definição, nuvens de pontos LiDAR, imagens de câmeras ou pose de pedestres como no OpenPose. Dessa forma, o trabalho considera tanto dados obtidos a partir da perspectiva BEV quanto conjuntos de dados obtidos a partir de um veículo, mas que posicionam as informações referentes a um sistema de coordenadas do mundo e não no plano da imagem. Essa abordagem torna o movimento dos agentes independente do movimento da câmera, o que faz o *Trajectron++* se distanciar de modelos como o PIE_{traj}.

Para levar em consideração diferentes agentes e as dinâmicas de cada um, as RNNs de saída do *Trajectron++* predizem os parâmetros de uma distribuição gaussiana de duas variáveis: posição e velocidade. Esses parâmetros são utilizados então pelo modelo de cada classe de agente para predizer a trajetória futura do nó em questão com base no seu modelo dinâmico.

Por utilizar um CVAE, o treinamento do *Trajectron++* envolve codificar informações anotadas da trajetória predita juntamente com o conjunto de observação, mas por um codificador separado. Durante o treinamento, é utilizada uma função de perda baseada na divergência de Kullback-Leibler, conforme revisado na seção 2.2.4, que permite ao modelo codificar as dependências entre X_{obs} e Y_{pred}. Como resultado, o *Trajectron++* se tornou o estado da arte nas tarefas de predição de trajetória quando foi proposto.

Posteriormente, Mangalam *et al.* (2020) abordam o problema de predição de trajetória com foco em modelar aspectos sociais entre os pedestres (o fato de uma trajetória influenciar a outra) e também considerar que as trajetórias são condicionadas pelo destino de cada indivíduo. Dessa forma, o modelo batizado de *PECNet* prediz o destino de cada agente e tenta recuperar a trajetória a partir dessa informação. Embora essa última ideia não seja nova na literatura e ter sido utilizada em diferentes abordagens, o trabalho de Mangalam *et al.* (2020) conseguiu combinar a predição condicionada ao destino com os aspectos sociais para atingir o estado da arte para os conjuntos de dados abordados.

Além das características citadas acima, o trabalho de Mangalam *et al.* (2020) segue

a tendência da literatura atual de abordar o problema de predição de trajetória de forma multimodal, aprendendo então uma distribuição de possíveis posições futuras para cada agente ao longo do tempo. Para essa abordagem, Mangalam *et al.* (2020), assim como Salzmann *et al.* (2020), baseiam seu modelo no uso de um CVAE.

A rede PECNet, proposta por Mangalam *et al.* (2020), codifica cada uma das trajetórias passadas de forma independente por meio de um codificador de trajetória passada e utiliza um CVAE para aprender a prever múltiplos pontos de destino (objetivo) a partir dessa codificação. Uma vez obtidos os destinos, as trajetórias são reconstruídas utilizando também aspectos de interação social entre os pedestres (trajetórias coexistentes temporalmente e suficientemente próximas interferem umas nas outras).

Todos os módulos propostos por Mangalam *et al.* (2020) são MLPs. Para treinar o modelo foi utilizada uma função de custo que combina a expressão da divergência de Kullback-Leibler para a variável latente (aprender as dependências entre X_{obs} e os objetivos verdadeiros) somada da distância quadrática média dos pontos de destino preditos e da distância média entre a trajetória predita e a real. Cada um desses termos da soma pode ainda ser ponderado por constantes a serem escolhidas pelo projetista.

Motivados pelo fato de que o problema de predição de trajetória de pedestres era abordado primordialmente por redes LSTM em que o progresso era feito apenas modelando a interação entre vários indivíduos na cena, Giuliari *et al.* (2021) propuseram a utilização de redes Transformadoras, que haviam superado as redes LSTM nas tarefas de processamento de linguagem natural, no contexto deste trabalho.

O método proposto simplesmente aplica o modelo de rede Transformadora clássico, ilustrado na Figura 6 e explicado na seção 2.2.3. Foram escolhidos, porém, 6 blocos empilhados no codificador e 8 no decodificador, sendo a entrada do modelo as trajetórias de cada um dos pedestres separadamente durante a janela de observação.

Apesar de possuir resultados piores que o modelo *Trajectron++*, de Salzmann *et al.* (2020), que já havia sido publicado, o modelo de Giuliari *et al.* (2021) superou todos os demais métodos do estado da arte no momento da publicação (omitidos neste documento), sejam eles baseados em análises individuais ou sociais dos pedestres. Os autores citam ainda que, pelo fato de o *Trajectron++* utilizar não só informações sociais, mas também de mapa, ele não deveria ser comparado ao modelo baseado unicamente em redes Transformadoras.

Yao *et al.* (2021) propuseram um novo modelo multimodal baseado em CVAEs chamado *BiTraP* (*Bi-Directional Pedestrian Trajectory Prediction*). Como contribuição, o trabalho de Yao *et al.* (2021) propõe um decodificador que oferece melhorias significativas em relação ao estado da arte, em especial para longos horizontes de predição (maiores que 2 segundos). Além disso, os autores foram os primeiros a avaliar o método proposto tanto

em conjuntos de dados a partir da perspectiva de um veículo quanto na perspectiva BEV.

Por fim, os autores avaliam ainda os impactos das duas possíveis formas de se modelar a variável latente de um CVAE: como uma distribuição categórica, como em Mangalam *et al.* (2020), ou como um modelo de mistura gaussiana (GMM) como em Salzmann *et al.* (2020). A principal implicação prática no modelo final diz respeito à forma da saída do modelo, uma vez que no caso categórico, são recuperadas as coordenadas das múltiplas trajetórias preditas enquanto que na abordagem utilizando o GMM, são recuperados parâmetros que descrevem a distribuição final e as coordenadas são encontradas a partir da integração sobre esses parâmetros.

Os autores adotam uma abordagem parecida com Giuliari *et al.* (2021) na medida em que as predições feitas não são diretamente a posição da trajetória e sim um deslocamento referente à posição anterior. Essa abordagem garante erros iniciais mais baixos do que estimar as posições absolutas (e consequentemente acelera a convergência), uma vez que se conhece a última posição observada e não são esperadas grandes mudanças a partir dessa posição para os primeiros instantes da trajetória.

O modelo *BiTraP* representou um avanço em relação ao PECNet e ao *Trajectron++* para o conjunto de dados abordado, se tornando o estado da arte no momento de sua publicação mesmo se baseando apenas nas trajetórias de entradas, sem utilizar nenhuma informação complexa da cena ou aspectos de interação social. Os autores analisaram ainda a versão da rede *BiTraP* utilizando GMM (*BiTraP-GMM*) ao invés da abordagem não paramétrica e concluíram que a *BiTraP-GMM* apresenta resultados inferiores à *BiTraP*.

A rede *BiTraP* representou uma melhora em relação aos métodos também em relação aos resultados obtidos a partir da perspectiva de uma câmera acoplada a um veículo, considerando ambas as versões multimodais. Além desses modelos, Yao *et al.* (2021) avaliaram também uma versão determinística da *BiTraP* (*BiTraP-D*), sem o CVAE e possuindo, portanto, uma saída unimodal. Ainda nessa versão, os resultados obtidos foram melhores que o estado da arte, comprovando a eficiência da abordagem de predizer os pontos de destino e reconstruir a trajetória a partir deles mesmo para câmeras móveis.

Mangalam *et al.* (2021) se distanciam um pouco da tendência da literatura recente de aplicar CVAEs e de tratar o problema de predição de trajetórias multimodais como um problema de regressão e propõem a Y-net, uma rede neural profunda de predição de trajetória baseada apenas no uso de CNNs. Esse modelo aborda a multimodalidade da predição modelando o problema como um problema de classificação, numa abordagem parecida com a de Ridel *et al.* (2020), mas adicionando a predição de objetivos futuros ao modelo e reconstruindo a trajetória final a partir deles.

Mangalam *et al.* (2021) dizem ainda modelar as incertezas envolvidas no processo de predição de trajetória dividindo ela em duas classes: incertezas epistêmicas, que são

conhecidas pelo agente, mas desconhecidas pelo modelo, e incertezas aleatórias, que são desconhecidas tanto pelo modelo quanto pelo agente. As primeiras são, por exemplo, objetivos de longo prazo, como o ponto final da trajetória futura, enquanto as demais são, por exemplo, a intenção de outros pedestres na cena e a aleatoriedade envolvida na decisão dos agentes.

Apesar de as definições remeterem àquelas utilizadas por Bhattacharyya, Fritz e Schiele (2018), no trabalho de Mangalam *et al.* (2021) não há nenhuma referência ao aprendizado bayesiano ou alguma forma de demonstração matemática de que as incertezas calculadas pelo modelo produzem uma estimativa real das incertezas envolvidas no processo. Sendo assim, apesar de a nomenclatura utilizada ser semelhante, o trabalho de Mangalam *et al.* (2021) não está relacionado à área de aprendizado profundo bayesiano.

A Y-net trata a incerteza epistêmica por meio da multimodalidade dos objetivos futuros, enquanto que a incerteza aleatória é modelada por meio da multimodalidade de trajetórias. Sendo assim, apesar da nomenclatura, a abordagem é bem semelhante às de Mangalam *et al.* (2020) e de Yao *et al.* (2021). Como resultado, o modelo é o estado da arte no momento presente para um conjunto de dados com imagens na perspectiva BEV.

Por fim, Mangalam *et al.* (2021) avaliaram ainda o desempenho da Y-net para horizontes de predição muito longos, com duração acima de $2,0\ s$. Para isso, utilizaram um conjunto de dados chamado *Intersection Drone Dataset*, focado em imagens aéreas e fixas de áreas de cruzamento. A partir desses dados, os autores concluíram que o aumento do número de pontos de objetivo intermediário no modelo e, consequentemente, do efeito da multimodalidade das trajetórias preditas, é benéfico para o modelo final.

O modelo conhecido como SGNet proposto por Wang *et al.* (2022) corresponde ao estado da arte atual para algumas métricas da tarefa de predição de trajetória de pedestres, tendo sido desenvolvido de forma invariante à perspectiva de imagem de entrada e tendo êxito tanto na tarefa de predição a partir de uma câmera fixa quanto a partir do ponto de vista de uma câmera acoplada a um veículo. Apesar de não fazer referência ao método de Mangalam *et al.* (2021), Wang *et al.* (2022) utilizaram a ideia de predizer não só o objetivo final, mas também vários objetivos intermediários, que são também atualizados ao longo do tempo, para recuperar a trajetória final a partir desses objetivos.

O SGNet é composto por três módulos: um estimador de objetivos a cada passo, um codificador que codifica os dados passados de trajetória e dos objetivos preditos em instantes anteriores e um decodificador que leva em consideração os objetivos intermediários preditos e aplica um mecanismo de atenção adaptativa para aprender a influência de cada objetivo predito e aprimorar o desempenho do modelo. Dentre eles, o estimador de objetivos a cada passo pode ser implementado por diversas arquiteturas diferentes (GRU, MLP ou CNN). Dentre os experimentos feitos, os autores mostraram que independente da escolha para esse módulo, a SGNet atinge o estado da arte para conjuntos de dados

obtidos a partir da perspectiva de um veículo.

Ainda, os autores mostraram também que, para conjuntos de dados dessa perspectiva, tanto a versão multimodal quando uma versão unimodal (SGNet-ED) atingem o estado da arte. É feita ainda uma análise que conclui que, para a SGNet, quanto maior o número de objetivos preditos ao longo da trajetória, melhor o resultado do modelo. Essa característica é semelhante à encontrada por Mangalam *et al.* (2021) para a Y-net.

Para um conjunto de dados de imagens na perspectiva BEV, apenas a SGNet-ED foi avaliada, sendo superada por pouco pela Y-net, que é multimodal e que não foi considerada por Wang *et al.* (2022) em seu trabalho. Dessa forma, uma comparação mais justa seria entre SGNet e Y-net e, dado o fato de os resultados da SGNet-ED serem próximos daqueles reportados pela Y-net, tudo leva a crer que a versão multimodal também se tornaria o estado da arte para esse tipo de entrada.

Posteriormente, Su *et al.* (2022) utilizaram a *Crossmodal Transformer* junto de um CVAE para combinar de forma mais complexa as informações multimodais de entrada (posição do pedestre, imagem de contexto, velocidade e orientação do veículo). O modelo também é treinado para predizer juntamente a intenção do pedestre de atravessar a via, como forma de auxiliar o treinamento da tarefa principal de predição de trajetória, tida como mais complexa. Ainda, ao invés de decodificar a trajetória diretamente, o modelo prediz três pontos de controle a partir dos quais é calculada uma curva de Berzier que representa a trajetória final.

Apesar de os autores também não se compararem com a SGNet, o trabalho mostrou resultados melhores que os da BiTraP para o conjunto de dados PIE. Em relação à SGNet, o modelo de Su *et al.* (2022) apresenta métricas melhores em horizontes longos de predição, principalmente pelo fato do modelo ter sido treinado com uma função de custo que aumenta exponencialmente o peso do erro quadrático médio ao longo da trajetória predita.

Por fim, Czech *et al.* (2023) propuseram o modelo determinístico *Behavior-Aware Pedestrian Trajectory Prediction* (BA-PTP) baseado em GRUs para codificar informações espaço-temporais do veículo em movimento por meio de sua trajetória estimada e informações do comportamento do pedestre, que no caso do conjunto de dados PIE é dada pela pose do pedestre, estimada por um modelo separado. O modelo alcançou o melhor desempenho quando todas as entradas descritas são utilizadas e supera a SGNet para métricas de predição mais longas quando é adicionado o módulo que codifica a trajetória estimada do veículo de entrada.

3.4 Considerações finais

A Tabela 1 apresenta um compilado dos estudos que apresentam modelos de predição de trajetória e que foram escolhidos para fazer parte da revisão deste trabalho.

Diversos outros modelos poderiam ter sido adicionados a essa revisão, como modelos para predição de trajetória de veículos, modelos desenvolvidos para câmeras de segurança ou modelos que atuam sobre mapas e predizem a trajetória de todos os agentes do trânsito.

A Tabela 2 mostra uma compilação dos resultados dos métodos revisados e que tiveram resultados avaliados nos conjuntos de dados PIE e JAAD. Nesses conjuntos de dados, que possuem imagens obtidas a partir da perspectiva de um veículo, o trabalho de Wang *et al.* (2022) corresponde ao estado da arte atual, tanto entre os métodos multimodais quanto para os métodos determinísticos.

Tabela 2 – Tabela de comparação entre os resultados reportados nos estudos de predição de trajetória analisados neste trabalho para os conjuntos de dados PIE e JAAD utilizando $0,5$ s de observação e $0,5/1,0/1,5$ s de horizonte de predição para o erro quadrático médio (MSE) e $0,5$ s para o erro quadrático médio considerando o centro do retângulo que representa o pedestre (C_{MSE}) e o C_{MSE} para o último instante de predição (C_{FDE}). Para métodos multimodais (*), as métricas retratadas dizem respeito à melhor predição dentre 20 dadas como saída. † Modelos que utilizam apenas a trajetória do pedestre como entrada.

	JAAD			PIE		
Modelo	MSE	C_{MSE}	C_{FDE}	MSE	C_{MSE}	C_{FDE}
LSTM Bayesiana	159/539/1535	1447	5615	101/296/855	811	3259
PIE_{traj}	110/399/1248	1183	4780	58/200/636	596	2477
HT-STGCN	-	-	-	-/-/551	-	-
Santos e Grassi (2021)	-	-	-	-/-/-	1105	4357
BiTraP*	38/94/222	177	565	23/48/102	81	261
BiTraP-GMM*	153/250/585	501	998	38/90/209	171	368
BiTraP-D	93/378/1206	1105	4565	41/161/511	481	1949
† SGNet*	37/86/197	146	443	16/39/88	66	206
† SGNet-ED	82/328/1049	996	4076	34/133/442	413	1761
Su *et al.* (2022)*	38/89/201	160	484	20/43/92	67	203
Su *et al.* (2022)	93/341/1026	979	3876	43/149/443	413	1670
BA-PTP completo	-	-	-	46/137/411	381	1593
† BA-PTP	-	-	-	53/188/615	580	2469

Fonte: Elaborado pelo autor.

Já as Tabelas 3 e 4 mostram os resultados dos modelos para imagens obtidas na perspectiva BEV utilizando os conjuntos de dados Stanford *drone* e ETH+UCY. Para a Tabela 3, a Y-net é o modelo de melhores resultados reportados. Apesar disso, a rede SGNet não foi avaliada nesse conjunto de dados. Já para o conjunto ETH+UCY, ambas as redes neurais foram avaliadas, de modo que a Y-net é a de melhor resultado, apesar de apenas a versão determinística da SGNet ter sido reportada.

Tabela 3 – Resultados dos métodos presentes neste texto para o conjunto de dados Stanford *drone*. São mostrados o número de trajetórias preditas (K) e reportadas as métricas de distância média do último ponto da trajetória predita e o valor real (FAD) e distância média entre cada um dos pontos da trajetória predita e a trajetória real (MAD) para a melhor das K trajetórias geradas por cada um dos modelos.

Modelo	K	FAD	MAD
Ridel *et al.* (2020)	5	$14,35$	$26,85$
PECNet	5	$12,79$	$25,98$
Y-net	5	$11,49$	$20,23$
PECNet	20	$9,96$	$15,88$
Y-net	20	$7,85$	$11,85$

Fonte: Elaborado pelo autor.

Tabela 4 – Resultados para o conjunto de dados ETH+UCY em função de cada uma da cenas do conjunto de dados e também a média geral sobre todo o conjunto. Métricas utilizadas: distância média entre a trajetória predita e a real/distância média entre o destino predito e o real. Resultados para abordagem determinística ou melhor resultado dentre 20 para o caso dos métodos multimodais (*). Para os experimentos, são utilizados 8 quadros de observação e 12 como horizonte de predição.

Modelo	ETH	Hotel	UCY	Zara2	Zara2	Média
Trajectron++ (*)	$0,50/1,19$	$0,24/0,59$	$0,36/0,89$	$0,29/0,72$	$0,27/0,67$	$0,34/0,76$
Transformadora	$1,03/2,10$	$0,36/0,71$	$0,53/1,32$	$0,44/1,00$	$0,34/0,76$	$0,54/1,17$
PECNet (*)	$0,54/0,87$	$0,18/0,24$	$0,35/0,60$	$0,22/0,39$	$0,17/0,30$	$0,29/0,48$
BiTraP (*)	$0,37/0,69$	$0,12/0,21$	$0,17/0,37$	$0,13/0,29$	$0,10/0,21$	$0,18/0,35$
BiTraPGMM (*)	$0,40/0,74$	$0,13/0,22$	$0,19/0,40$	$0,14/0,28$	$0,11/0,22$	$0,19/0,37$
Y-net (*)	$0,28/0,33$	$0,10/0,14$	$0,24/0,41$	$0,17/0,27$	$0,13/0,22$	$0,18/0,27$
SGNet-ED	$0,35/0,65$	$0,12/0,24$	$0,20/0,42$	$0,12/0,24$	$0,10/0,21$	$0,18/0,35$

Fonte: Elaborado pelo autor.

4 METODOLOGIA

Conforme apresentado nos capítulos anteriores, o campo de predição de trajetória de pedestres para veículos autônomos é dominado por redes neurais geradoras, em especial por arquiteturas baseadas em CVAEs. Entretanto, esses modelos são treinados para minimizar a melhor dentre múltiplas trajetórias possíveis obtidas a partir da amostragem da FDP de saída, buscando obter saídas probabilísticas mais diversas (MANGALAM *et al.*, 2021), o que ocorre também para métodos baseados em grades (RIDEL *et al.*, 2020).

O trabalho de Bhattacharyya, Fritz e Schiele (2018) é o único do meio a modelar a incerteza epistêmica heterocedástica dentro deste campo de pesquisa, de modo que, para modelos que seguem a abordagem Bayesiana, o objetivo central é obter uma predição precisa seguida de uma estimativa de incerteza para aquela medida dada a arquitetura da rede. Dessa forma, o objetivo de obtenção da FDP de saída é diferente daquele observado em redes geradoras e as métricas utilizadas para avaliação do trabalho de Bhattacharyya, Fritz e Schiele (2018) seguem as padrões reportadas para os modelos determinísticos.

Este trabalho, busca então, propor uma metodologia para treinar uma rede neural de aprendizado profundo para predição de trajetória de pedestres que modele a incerteza preditiva multivariada do modelo. Essa incerteza multivariada é composta pelas componentes heterocedásticas aleatórias e epistêmicas, conforme o trabalho de Russell e Reale (2021), aplicado aos problemas de odometria visual e de rastreamento.

Além de aplicar uma metodologia diferente da proposta por Bhattacharyya, Fritz e Schiele (2018) para modelagem da incerteza do modelo, a escolha por tomar o trabalho de Russell e Reale (2021) como base trás ainda a capacidade de se obter estimativas multivariadas. Como contribuição em relação à LSTM Bayesiana, este trabalho mostra como adicionar relações matemáticas necessárias ao trabalho de Russell e Reale (2021) e como garantir estabilidade numérica durante o treinamento do modelo.

4.1 Estimativa multivariada de incerteza

4.1.1 Incerteza aleatória heterocedástica

Conforme apresentado na Seção 2.3.1, esta componente é aprendida pelo modelo de rede neural por meio da minimização da função de custo durante o treinamento. Entretanto, dado que $\Sigma(x)$ é uma saída do modelo a ser treinado, é preciso impor condições sobre o modelo para que a Equação 2.27 seja sempre válida, independente do número de variáveis do problema.

4.1.2 Garantia de estabilidade numérica

Matrizes de covariância são semi-definidas positivas, o que implica em $|\Sigma(x)| \geq 0$. Entretanto, para $|\Sigma(x)| = 0$, os termos $\Sigma^{-1}(x)$ e $\log|\Sigma(x)|$ da Equação 2.27 não são definidos. Além disso, do ponto de vista numérico, valores de determinante próximos a 0 podem trazer instabilidades ao cálculo da matriz inversa e do logaritmo desse número natural. Dessa forma, para que o treinamento de um modelo aconteça, é necessário garantir que a matriz de covariância predita seja definida positiva ($|\Sigma(x)| > 0$).

4.1.2.1 Decomposição de Cholesky

A decomposição de Cholesky se baseia em, dada uma matriz A qualquer, determinar uma matriz triangular $\hat{L}$ para a qual vale $A = \hat{L} \cdot \hat{L}^T$. Neste caso, segue ainda que $det(A) = (det(\hat{L}))^2$. Dessa forma, no contexto de uma rede neural que prediz uma matriz de covariância que precisa ser definida positiva, a estratégia a ser adotada consiste em montar $\hat{L}$ a partir da saída do modelo de modo que o produto dos termos ao longo da diagonal principal (determinante de uma matriz triangular) seja sempre positivo.

Considerando um problema hipotético com quatro variáveis de entrada (u, v, w, h), a matriz de covariâncias do modelo é descrita por:

$$\Sigma(x) = \begin{bmatrix} \sigma_u^2 & cov(u,v) & cov(u,w) & cov(u,h) \\ cov(v,u) & \sigma_v^2 & cov(v,w) & cov(v,h) \\ cov(w,u) & cov(w,v) & \sigma_w^2 & cov(w,h) \\ cov(h,u) & cov(h,v) & cov(h,w) & \sigma_h^2 \end{bmatrix}. \tag{4.1}$$

Uma rede neural artificial é então treinada para predizer os conjuntos de coeficientes $S = \{s_0, ..., s_n\}$ e $R = \{r_{21}, ..., r_{ij}\}$, para $i = \{2, ..., n\}$, $i > j$ e n variáveis de entrada na equação:

$$\hat{L} = \begin{bmatrix} e^{s_0} & 0 & 0 & 0 \\ r_{21} & e^{s_1} & 0 & 0 \\ r_{31} & r_{32} & e^{s_2} & 0 \\ r_{41} & r_{42} & r_{43} & e^{s_3} \end{bmatrix}, \tag{4.2}$$

na qual termos exponenciais dispostas ao longo da diagonal de $\hat{L}$ buscam possibilitar que as variâncias preditas sejam reduzidas a valores muito baixos.

A grande questão por trás do uso da decomposição de Cholesky com redes neurais artificiais para a predição de matrizes de covariâncias está no fato de a expressão de alguns dos termos de covariância obtidos a partir de $\hat{L}$ não estar relacionada às demais covariâncias que envolvem o par de variáveis que ela representa. Exemplificando, na Equação 4.3 o processo de treinamento do modelo pode otimizar $\sigma_w^2 = r_{31}^2 + r_{32}^2 + e^{2s_2}$ reduzindo o

termo s_2, o que implica em mudanças em $cov(h, w) = r_{31}r_{41} + r_{32}r_{42} + r_{43}e^{s_2}$, mas não em $cov(u, w) = r_{31}e^{s_0}$ e $cov(v, w) = r_{21}r_{31} + r_{32}e^{s_1}$.

$$\Sigma(x) = \begin{bmatrix} e^{2s_0} & r_{21}e^{s_0} & r_{31}e^{s_0} & r_{41}e^{s_0} \\ r_{21}e^{s_0} & r_{21}^2 + e^{2s_1} & r_{21}r_{31} + r_{32}e^{s_1} & r_{21}r_{41} + r_{42}e^{s_1} \\ r_{31}e^{s_0} & r_{21}r_{31} + r_{32}e^{s_1} & r_{31}^2 + r_{32}^2 + e^{2s_2} & r_{31}r_{41} + r_{32}r_{42} + r_{43}e^{s_2} \\ r_{41}e_{s_0} & r_{21}r_{32} + r_{42}e^{s_1} & r_{31}r_{41} + r_{32}r_{42} + r_{43}e^{s_2} & r_{41}^2 + r_{42}^2 + r_{43}^2 + e^{2s_3} \end{bmatrix} \tag{4.3}$$

4.1.2.2 Modelagem multivariada explícita

Ao invés de predizer coeficientes que formam uma forma de decomposição da matriz de covariância, a modelagem multivariada explícita busca que a rede neural estime o coeficiente de correlação de Pearson (CCP) para cada par de variáveis de entrada. Seja ρ_{de} o CCP entre variáveis arbitrárias d e e. É possível então definir a covariância em função do CCP e do desvio padrão de cada variável (BENESTY *et al.*, 2009) conforme a seguinte equação:

$$\rho_{de} = \frac{cov(d, e)}{\sigma_d \sigma_e} \longrightarrow cov(d, e) = \rho_{de}\sigma_d\sigma_e. \tag{4.4}$$

Russell e Reale (2021) definiram os conjuntos de coeficientes S (termos ao longo da diagonal) e R (termos abaixo da diagonal) conforme pelas equações abaixo:

$$\Sigma_{ii}(x) = e^{s_i}, \tag{4.5}$$

$$\Sigma_{ij}(x) = \Sigma_{ji}(x) = \tanh(r_{ij})\sqrt{e^{s_i} \cdot e^{s_j}}. \tag{4.6}$$

Dessa forma, do ponto de vista da rede neural, o conjunto S é definido por uma camada com função de ativação linear, enquanto que sobre o conjunto R é aplicada uma função de ativação tangente hiperbólica. Além disso, a redução de um valor de variância dado pela Equação 4.5, implica diretamente na redução do valor de todas as covariâncias relacionadas àquela variável.

Do ponto de vista da inversibilidade da matriz de covariância, para o problema com duas variáveis de entrada, a seguinte equação mostra que $\Sigma(x)$ é definida positiva:

$$|\Sigma_{2\times2}(x)| = (1 - \tanh^2(r_0)) \cdot e^{s_0+s_1}. \tag{4.7}$$

Entretanto, para um problema com 3 variáveis arbitrárias u, v e w, a expressão do determinante da matriz de covariâncias é dada por:

$$|\Sigma_{3\times3}(x)| = \sigma_u^2\sigma_v^2\sigma_w^2[1 + 2\rho_{uv}\rho_{vw}\rho_{uw} - \rho_{uw}^2 - \rho_{vw}^2 - \rho_{uv}^2]. \tag{4.8}$$

Ao tratar os CCPs de forma independente conforme proposto por Russell e Reale (2021) na Equação 4.8 é possível que o modelo infira $r_{21},\ r_{31},\ r_{32}$ tais que $\rho_{uv} = \rho_{uw} = \rho_{vw} = -1$, levando a $|\Sigma_{3x3}(x)| = -4\sigma_u^2\sigma_v^2\sigma_w^2$.

O trabalho de Russell e Reale (2021) deixa de modelar a relação de proporcionalidade que existe entre pares de variáveis quando analisados dois a dois. Por exemplo, para o caso tridimensional, $\rho_{uv} < 0$ e $\rho_{uw} < 0$ indicam que v e w variam em relação oposta a u. Dessa forma, v e w obrigatoriamente variam na mesma direção, o que implica em $\rho_{vw} > 0$.

Para que a matriz de covariância para 3 variáveis de entrada seja então definida positiva, com base na Equação 4.8 chega-se na condição descrita pela seguinte equação:

$$1 + 2\rho_{uv}\rho_{vw}\rho_{uw} - \rho_{uw}^2 - \rho_{vw}^2 - \rho_{uv}^2 > 0. \tag{4.9}$$

Definindo $a = \rho_{uv}\rho_{vw}$ e $c = 1 - \rho_{vw}^2 - \rho_{uv}^2$, basta então que o modelo faça a predição de ρ_{uv} e de ρ_{vw} e que ρ_{uw} seja definido de modo a respeitar a condição descrita por:

$$a - \sqrt{a^2 + c} < \rho_{uw} < a + \sqrt{a^2 + c}. \tag{4.10}$$

Por exemplo, definir $\rho_{uw} = \rho_{uv}\rho_{vw}$ não só garante a inversibilidade da matriz de covariância como também maximiza a expressão do determinante, o que auxilia na estabilidade numérica. Esta formulação não foi proposta por Russell e Reale (2021) e consiste numa contribuição deste trabalho.

Além da relação de variação entre pares de covariâncias, outra adição importante ao trabalho de Russell e Reale (2021) consiste em garantir que o valor mínimo admissível do determinante da matriz de covariância seja bem definido por um número de ponto flutuante de 32 *bits*. O determinante da matriz de covariância pode ser definido pelo produto de seus autovalores (λ), conforme mostrado na seguinte equação:

$$|\Sigma(x)| = \prod_{i=1}^{n} \lambda_i. \tag{4.11}$$

Cada um dos autovalores pode ser determinado ao calcular as raízes do polinômio característico da matriz de covariância:

$$p(\lambda) = det(\lambda I - \Sigma(x)). \tag{4.12}$$

Para garantir que o determinante seja sempre bem representado em 32 *bits*, define-se um valor ϵ para o qual a matriz de covariância final $\Sigma'(x) = \Sigma(x) + \epsilon I$ de modo que a

expressão do polinômio característico se torne $p(\lambda') = det((\lambda - \epsilon)I - \Sigma(x))$. Sendo assim, basta definir ϵ em função do valor mínimo de número de ponto flutuante F_{min} e do número de variáveis de entrada, conforme a equação:

$$\epsilon >= \frac{F_{min}}{n}. \tag{4.13}$$

No caso do conjunto de dados PIE, que envolve predizer quatro variáveis de saída, definir $\epsilon = 1,0 \cdot 10^{-7}$ é suficiente para satisfazer a Equação 4.13 sem atrapalhar a precisão das predições. Como as imagens de entrada têm dimensão igual a 1920×1080 e considerando covariâncias preditas para posições normalizadas no intervalo $[0, 1]$, desnomalizar a $i-$ésima variância dada por $e^{s_i} + \epsilon$ considerando $e^{s_i} \longrightarrow 0$, resulta em $\sigma_i^2 = (1920)^2 \cdot \epsilon$, que é menos que 1 pixel.

4.1.3 Incerteza epistêmica heterocedástica

A incerteza epistêmica multivariada heterocedástica é calculada pela Equação 2.30 conforme detalhado na Seção 2.3.3. À soma da incerteza epistêmica (Σ^{epi}) com a incerteza aleatória (Σ^{ale}) é dado o nome de incerteza preditiva (Σ^{pred}). Dessa forma, a incerteza preditiva é descrita pela seguinte equação:

$$\begin{aligned} &\Sigma^{pred} \approx \Sigma^{epi} + \Sigma^{ale}, \\ &\Sigma^{pred} \approx \frac{1}{N}\sum_{n=1}^{N} f_n(x) f_n(x)^T - \frac{1}{N^2}(\sum_{n=1}^{N} f_n(x))(\sum_{n=1}^{N} f_n(x))^T + \frac{1}{N}\sum_{n=1}^{N} \Sigma_n(x). \end{aligned} \tag{4.14}$$

Independentemente do método de amostragem escolhido (Monte Carlo *Dropout* ou *Deep Ensembling*), Russell e Reale (2021) pontuam que a Equação 4.14 não precisa ser usada durante o treinamento se a incerteza epistêmica for pequena em relação à aleatória.

4.2 Experimentos

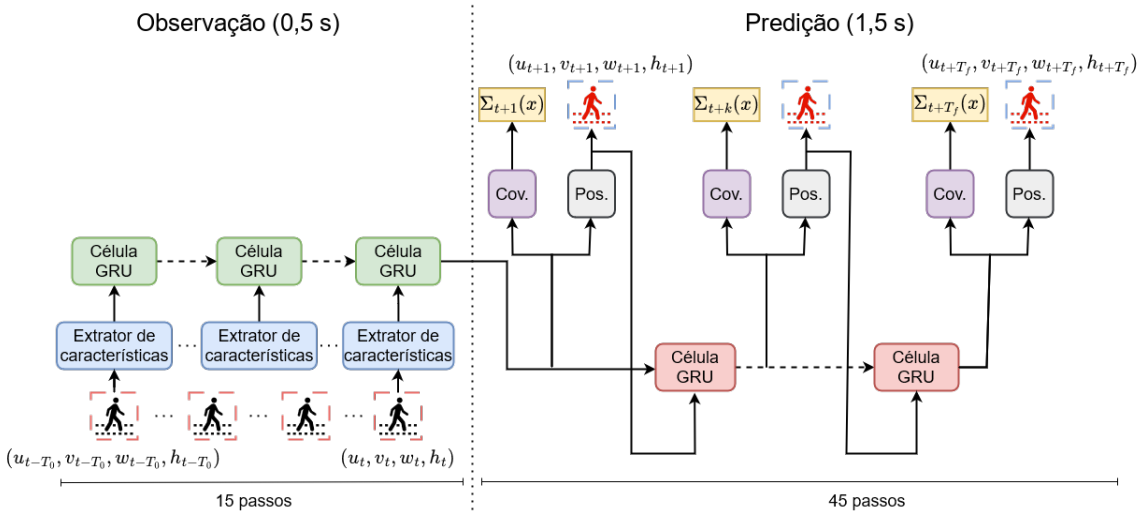
O conjunto de dados PIE proposto por Rasouli *et al.* (2019) foi escolhido para este trabalho. Ele contém 1842 trajetórias de pedestres capturadas por câmeras acopladas a um veículo que filmam na resolução 1920×1080 a uma frequência de 30 quadros por segundo. As trajetórias são amostradas com uma sobreposição de 50% e com $2,0$ segundos de duração. Os dados são divididos entre conjuntos de treino, validação e teste seguindo uma proporção de 50%, 10% e 40% respectivamente. Os $2,0$ s de duração da trajetória são divididos de forma que os primeiros $0,5$ s correspondem à entrada dos modelos e os $1,5$ s posteriores são os rótulos a serem preditos.

4.2.1 Modelos utilizados

Duas arquiteturas de redes neurais diferentes foram utilizadas nos experimentos. A primeira ilustrada na Figura 15 é um modelo simples baseado em realimentação composto por uma camada de extração de características utilizando a função de ativação ReLU (azul), um codificador (verde), um decodificador (vermelho) e duas camadas lineares para produzir as coordenadas de saída (cinza) e as matrizes de covariância (roxo). Tanto o codificador quando o decodificador são compostos por células de GRUs. Todas as camadas extraem características escondidas de dimensão igual a 512.

A retroalimentação permite que as células do decodificador realizem predições de posições futuras com base na última posição predita. As posições e as respectivas matrizes de covariância são decodificadas por camadas lineares, sendo que a camada de regressão das posições utiliza a função de ativação sigmoid.

Figura 15 – Diagrama do modelo proposto para predizer trajetórias futuras de pedestres e a matriz de covariância correspondente a cada predição ao longo do tempo.



Fonte: Elaborada pelo autor.

A metodologia proposta também foi aplicada à versão determinística do modelo SGNet (WANG *et al.*, 2022), que é baseado em células GRU que calculam trajetórias futuras a cada instante de tempo para aprender a estimar melhores predições ao final do conjunto de observações. Para adequar o modelo à metodologia proposta, a camada de regressão das matrizes de covariância mostrada na Figura 15 foi adicionada sobre o vetor de características produzido pela SGNet. Isso pode ser aplicado a qualquer modelo de rede neural de aprendizado profundo com mínimo impacto na arquitetura final.

4.2.2 Métricas utilizadas

Para avaliar os resultados das trajetórias preditas, foram calculadas três métricas utilizadas na literatura: o erro quadrático médio sobre as coordenadas que delimitam o pedestre no plano da imagem (MSE), a média do erro de deslocamento da coordenada central do pedestre (C_{MSE}) ao longo da trajetória e o mesmo erro de deslocamento para o último instante predito (C_{FDE}). Em relação ao MSE, os valores são calculados sobre as predições feitas para os $0,5$ s, $1,0$ s e para a sequência completa de $1,5$ s.

Já para avaliar a qualidade da incerteza estimada pelo modelo da Figura 15 após o treinamento com a metodologia proposta, foram adotados os procedimentos utilizados por Bhattacharyya, Fritz e Schiele (2018) e Ilg *et al.* (2018), que consistem em formas diferentes de visualizar a relação existente entre o valor de incerteza e do erro associado a uma predição.

Bhattacharyya, Fritz e Schiele (2018) propuseram calcular histogramas bidirecionais que mostram a relação entre cada tipo de incerteza (epistêmica, aleatória e preditiva) varia em relação ao erro quadrático médio de cada ponto predito no conjunto de testes. Dessa forma, nesse tipo de visualização, busca-se encontrar que a incerteza e o erro das amostras variam na mesma direção e sentido, ou seja, pontos com maior erro são aqueles cuja incerteza preditiva é maior. Além disso, é possível visualizar a ordem de grandeza da incerteza, do erro quadrático médio e a contagem de amostras das saídas obtidas para essas duas grandezas.

Ilg *et al.* (2018) propuseram calcular a curva de esparsificação para modelos de rede neural que estimavam o fluxo óptico de uma sequência de imagens de entrada. Esse gráfico mostra como o erro quadrático médio (MSE) varia a medida que a fração de predições que possui maior valor de incerteza preditiva é removida ao longo de cada trajetória do conjunto de testes. O gráfico do modelo proposto é comparado com a curva do oráculo, representa a variação do erro ao longo do conjunto de testes quando é realmente removido o ponto da trajetória que possui maior erro quadrático médio.

4.2.3 Configuração do ambiente

Os experimentos foram realizados utilizando um computador com o sistema operacional Ubuntu equipado com um processador Intel Core i7-7700K e uma placa de vídeo NVIDIA Titan X. Os modelos treinados foram desenvolvidos utilizando Python e a biblioteca PyTorch. Para a SGNet, foi utilizado o código fonte original do modelo disponibilizado pelos autores do artigo em repositório aberto [1]. Dessa forma, os parâmetros de treinamento desse modelo foram mantidos os mesmos do artigo original.

Já o modelo da Figura 15 foi treinado por 80 épocas utilizando 128 trajetórias por

[1] https://github.com/ChuhuaW/SGNet.pytorch

lote e o otimizador Adam com uma taxa de aprendizado igual a $0,01$ que é reduzido para 20% a cada cinco épocas sem melhora na função de perda calculada sobre o conjunto de validação.

Para avaliar a incerteza epistêmica, o método de amostragem escolhido foi o de criar um *deep ensemble*. Para isso, os modelos para predição de posições absolutas foram treinados 50 vezes, cada vez com uma semente única para inicialização dos pesos do modelo.

5 RESULTADOS E DISCUSSÕES

Dada a metodologia proposta no Capítulo 4, neste capítulo são mostrados os resultados quantitativos e qualitativos obtidos para os experimentos realizados no conjunto de dados PIE. Além disso, a medida que os resultados são apresentados, são feitas comparações com outros métodos do estado da arte e discussões a respeito da metodologia proposta.

5.1 Resultados quantitativos

A Tabela 5 mostra uma comparação entre o modelo da Figura 15 e outros métodos existentes na literatura. Foram reportadas métricas para o modelo treinado para predizer dois tipos diferentes de informação: a posição absoluta e o deslocamento entre cada passo k e $k-1$ ao longo da trajetória. Para ambos os casos, os dados de entrada e a arquitetura utilizada são as mesmas (tirando a mudança na função de ativação para tanh).

A literatura indica que predizer deslocamentos a cada instante de tempo leva a uma convergência mais rápida e melhores resultados (YAO *et al.*, 2021; WANG *et al.*, 2022). Esse mesmo comportamento pode ser observado na Tabela 5, entretanto, para recuperar as posições preditas dada a última posição observada e os deslocamentos preditos, é necessário acumular as matrizes de covariância afim de obter a covariância final. Essa operação resulta em valores mais altos de incerteza, conforme observado nos valores de desvio padrão σ_u e σ_v (referentes às médias das coordenadas que delimitam o centro do pedestre na imagem ao longo do horizonte de predição de 1.5 s) mostrados na Tabela 5.

A Tabela 5 também mostra que o estado da arte é dominado por redes neurais geradoras que predizem diversas possíveis trajetórias com o intuito de maximizar a chance de gerarem uma única predição precisa. Apesar disso, dentre os modelos determinísticos, a rede neural da Figura 15 supera trabalhos que exploram arquiteturas e entradas complexas, como em Santos e Grassi (2021), Bhattacharyya, Fritz e Schiele (2018), Rasouli *et al.* (2019), Yao *et al.* (2019) e em Czech *et al.* (2023) (quando considerada apenas a trajetória passada do pedestre). Indo além, o modelo apresenta uma melhora significativa na predição de trajetórias futuras quando comparado com a LSTM Bayesiana, que era o único método capaz de estimar a incerteza preditiva.

Em termos de custo computacional, o modelo da Figura 15 representa uma melhoria em relação à SGNet tanto em termos de memória quanto em tempo de execução. O modelo possui aproximadamente 2,38 milhões de parâmetros e complexidade $O(T_0+T_f)$ em relação às trajetórias de entrada, enquanto que a SGNet possui aproximadamente 4,36 milhões de parâmetros e complexidade $O(T_0 \cdot T_f)$.

Tabela 5 – Resultados determinísticos e multimodais para o conjunto de dados PIE em pixels.

Modelo	MSE	C_{MSE}	C_{FDE}	σ_u	σ_v
SGNet(WANG *et al.*, 2022)*	**16**/**39**/**88**	**66**	206	–	–
Su *et al.* (2022)*	20/43/92	67	**203**	–	–
BiTraP(YAO *et al.*, 2021)*	23/48/102	81	261	–	–
SGNet(WANG *et al.*, 2022)	**34**/**133**/442	413	1761	–	–
BA-PTP completo(CZECH *et al.*, 2023)	46/137/**411**	**381**	**1593**	–	–
BA-PTP†	53/188/615	580	2469	-	-
Su *et al.* (2022)	43/149/443	413	1670	–	–
BiTraP(YAO *et al.*, 2021)	41/161/511	481	1949	–	–
PIETraj(RASOULI *et al.*, 2019)	58/200/636	596	2477	–	–
FOL-X††(YAO *et al.*, 2019)	47/183/584	546	2303	–	–
BLSTM†††(BHATTACHARYYA; FRITZ; SCHIELE, 2018)	101/296/855	811	3259	–	–
Santos e Grassi (2021)	–/–/–	1105	4357	–	–
Modelo de retroalimentação[1](*média*)	44/178/572	542	2257	33	6.6
Modelo de retroalimentação[1](*melhor*)	41/162/527	498	2103	30	6.6
Modelo de retroalimentação[2](*média*)	43/176/570	541	2246	220	44.6
Modelo de retroalimentação[2](*melhor*)	40/160/513	484	2007	206	43.2
SGNet [1](*média*)	51/174/546	510	2164	1018	44.6
SGNet [1](*melhor*)	39/150/487	456	1950	911	42.7

*melhor resultado dentre 20 amostras. †Apenas a trajetória do pedestre como entrada. †††Reportado por Rasouli *et al.* (2019). ††Reportado por Yao *et al.* (2021).
[1]posição. [2]deslocamento.

Fonte: Elaborado pelo autor.

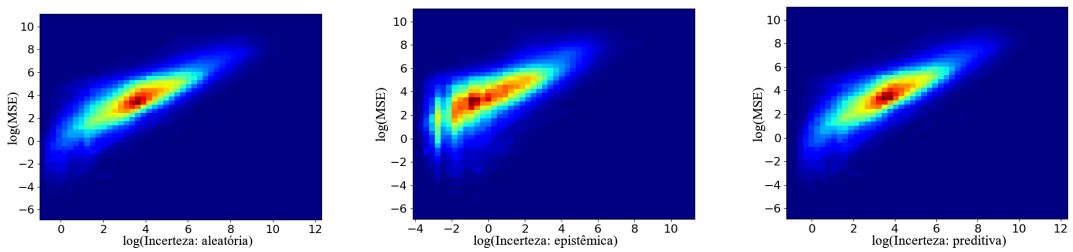
Essas propriedades da SGNet tornam difícil de aplicar métodos de amostragem como o *Deep Ensembling* e o *Monte Carlo Dropout* para calcular a incerteza epistêmica, já que a SGNet realiza inferências a uma taxa de 3,5 trajetórias por segundo (tps), enquanto o modelo da Figura 15 tem uma taxa de 28,3 tps. Dessa forma, sem computar a incerteza epistêmica realizando 50 amostras, é possível embarcar o modelo em um veículo autônomo considerando a GPU utilizada.

5.2 Validação da incerteza

A Figura 16 mostra histogramas bidirecionais que ilustram como o erro quadrátrico médio (MSE) é bem correlato à incerteza estimada, já que ambas as grandezas tendem a variar no mesmo sentido nos gráficos mostrados. Além disso, a incerteza epistêmica predita é menor que a incerteza aleatória para o mesmo valor de erro, o que é uma condição necessária para utilizar a Equação 4.14 apenas durante a inferência.

A Figura 17 mostra a curva de esparsificação para o modelo treinado. É possível

Figura 16 – Histogramas bidimensionais do logaritmo do erro quadrático médio em cada ponto de predição e em função do logaritmo de cada componente de incerteza estimado (considerando apenas os elementos ao longo da diagonal da matriz de covariâncias). As incertezas são bem correlacionadas ao erro já que valores altos do erro levam a valores altos de incerteza (e vice-versa).



Fonte: Elaborada pelo autor.

notar que o erro quadrático médio cai à medida que maiores frações de pontos são removidos, praticamente seguindo a curva do oráculo. Isso demonstra que a metodologia proposta para predição da incerteza é capaz de capturar corretamente o erro do modelo ao longo de cada trajetória.

Por fim, a Tabela 6 mostra a porcentagem de predições do conjunto de testes que estão distantes de um dado número de desvios padrão ($\#\sigma$) da predição realizada para cada componente de incerteza modelada, bem como para a incerteza preditiva. Apesar de os resultados não seguirem a regra $68-95-99.7$ de FDPs gaussianas (o que é esperado dado que o conjunto de dados não é balanceado), a Tabela 6 mostra que os dados correspondem à fonte de incerteza mais relevante para o modelo proposto nesta aplicação.

Tabela 6 – Percentual de predições distantes até um dado número de desvios padrões para o conjunto de testes

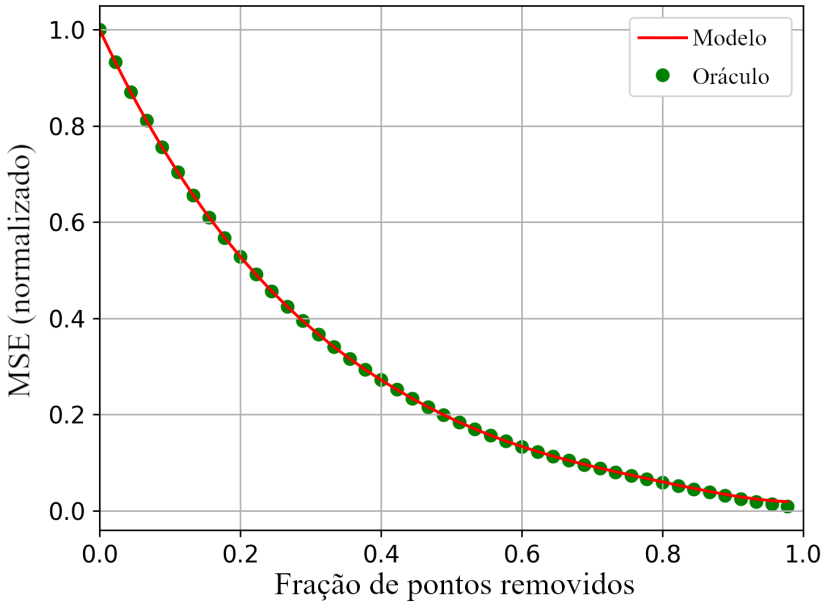
$\#\ \sigma$	Incerteza epistêmica (%)	Incerteza aleatória (%)	Incerteza preditiva (%)
1	11.81	86.02	92.05
2	32.05	98.40	99.28
3	50.98	99.61	99.84

Fonte: Elaborado pelo autor.

5.3 Resultados qualitativos

A Figura 18 mostra dois exemplos de sequências preditas pela metodologia proposta com as respectivas estimativas de incerteza preditiva para o conjunto de testes. A incerteza é mostrada pelos retângulos de diferentes cores, adicionando um, dois e três desvios padrão à predição realizada.

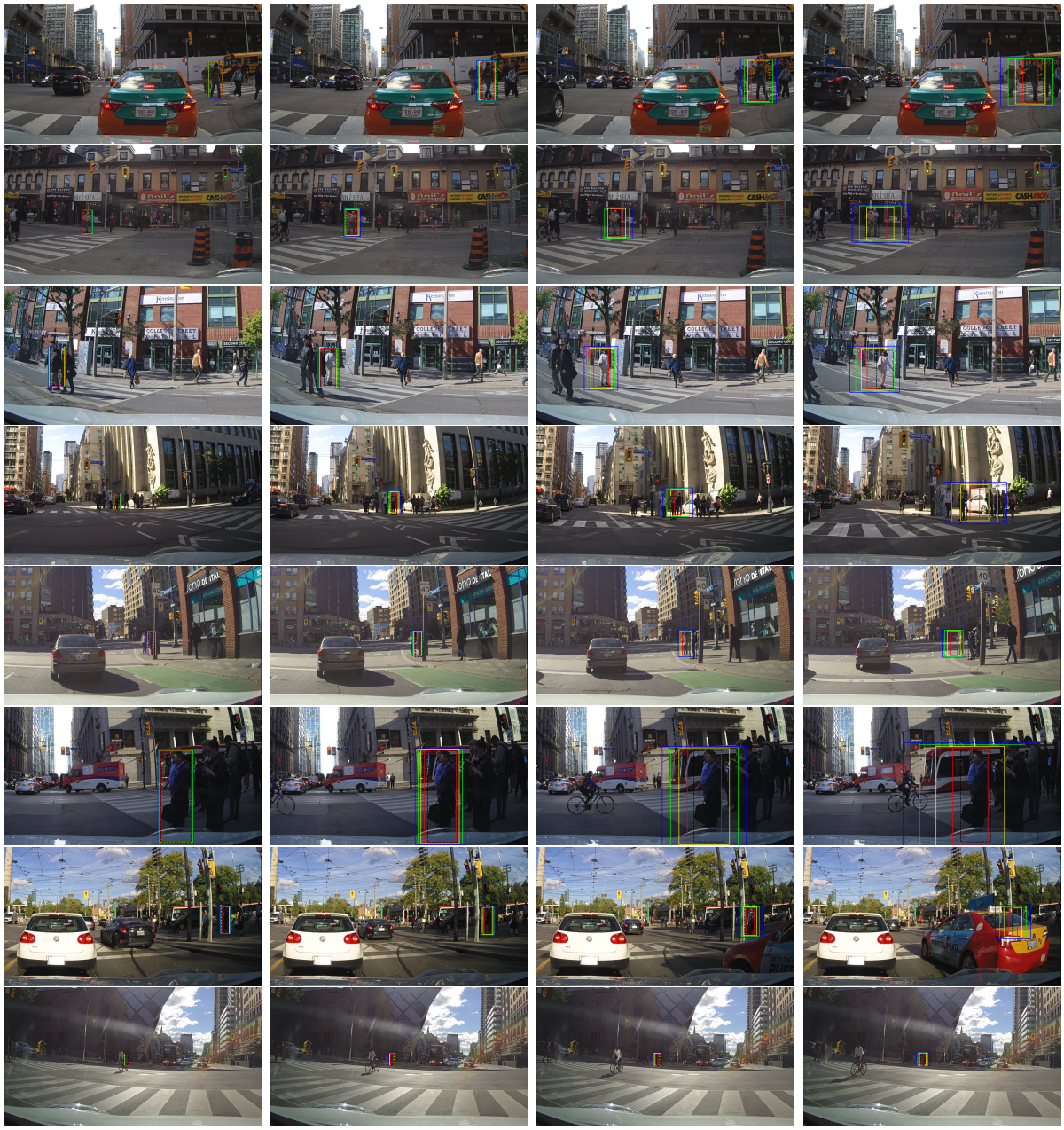
Figura 17 – Curva de esparsificação do modelo proposto para o conjunto de testes. Ela mostra como o erro quadrático médio varia à medida em que os pontos de incerteza mais alta são removidos de cada trajetória. A curva do oráculo mostra o limite inferior da variação do erro, que corresponde a remover os pontos de erro mais alto em relação à anotação.



Fonte: Elaborada pelo autor.

Pelos exemplos dados, é possível notar que a rede neural prediz trajetórias satisfatórias mesmo diante de situações complexas, como o pedestre atravessando a rua na frente de um carro em movimento. Além disso, a incerteza predita consegue descrever bem a região da imagem em que o pedestre está localizado mesmo quando a predição feita não se mostra precisa.

Figura 18 – Oito exemplos de trajetórias preditas para o conjunto de testes. Da direita para a esquerda, linha a linha: 1º/15º/30º/45º passo predito. As cores dos retângulos representam: vermelho - predição, preto (pontilhado) - valor real, amarelo/verde/azul - predições considerando 1/2/3 desvios padrão para as coordenadas do ponto central



Fonte: Elaborada pelo autor.

6 CONCLUSÃO

Este trabalho apresenta uma metodologia que permite treinar modelos de rede neural de aprendizado profundo para predizer a incerteza preditiva multivariada considerando as dependências entre os Coeficientes de Correlação de Pearson (CCPs) presentes na matriz de covariância. Para isso, este trabalho se baseia na metodologia proposta por Russell e Reale (2021) para predição de incerteza preditiva multivariada por modelos de aprendizado profundo, inicialmente aplicada aos problemas de rastreamento e de odometria visual. Ao trabalho de Russell e Reale (2021) foram adicionadas condições para modelar as relações de variação existentes entre os CCPs, que contribuem para a estabilidade numérica do treinamento.

É apresentada ainda uma comparação entre a metodologia proposta de modelar os CCPs com a saída da rede neural e a abordagem de predizer matrizes de covariância a partir da decomposição de Cholesky, outra abordagem presente na literatura. Entretanto, é discutida como a decomposição de Cholesky não se mostra adequada pelo fato de não modelar corretamente a relação entre os CCPs e de dificultar a diminuição dos termos preditos durante o treinamento. Além disso, à modelagem matemática proposta, é adicionada e discutida uma condição adicional que garante estabilidade numérica sem prejudicar a precisão das predições realizadas.

Por fim, este trabalho utiliza a metologia proposta para realizar a primeira aplicação da estimativa de incerteza multivariada em modelos de aprendizado profundo no campo de predição de trajetória de pedestres para veículos autônomos. A metodologia proposta é genérica e pode ser aplicada a qualquer arquitetura de rede neural com adaptações mínimas nas camadas do modelo. Em termos de resultados, um modelo de realimentação proposto baseado unicamente nas trajetórias de entrada supera a arquitetura LSTM Bayesiana, que é o único modelo do campo de estudo capaz de estimar algum tipo de incerteza de suas predições.

Em termos de possíveis continuações para o trabalho, um primeiro ponto interessante consiste em identificar a FDP que melhor representa o conjunto de dados e remodelar as equações utilizadas para estimativa de incerteza, que neste trabalho se baseiam em funções gaussianas. Além disso, é possível explorar outros tipos de entrada e de saída disponíveis no conjunto de dados endereçado com o intuito de modelar, por exemplo, o movimento do veículo e a intenção do pedestre. Por fim, é interessante avaliar e propor formas de relacionar matematicamente em uma nova arquitetura proposta a matriz de covariâncias de um instante com a matriz de covariância predita no instante anterior durante a etapa de decodificação do modelo.

REFERÊNCIAS

ABDAR, M.; POURPANAH, F.; HUSSAIN, S.; REZAZADEGAN, D.; LIU, L.; GHAVAMZADEH, M.; FIEGUTH, P.; CAO, X.; KHOSRAVI, A.; ACHARYA, U. R. *et al.* A review of uncertainty quantification in deep learning: Techniques, applications and challenges. **Information Fusion**, Elsevier, v. 76, p. 243–297, 2021.

ALAHI, A.; GOEL, K.; RAMANATHAN, V.; ROBICQUET, A.; FEI-FEI, L.; SAVARESE, S. Social lstm: Human trajectory prediction in crowded spaces. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2016. p. 961–971.

ALOM, M. Z.; TAHA, T. M.; YAKOPCIC, C.; WESTBERG, S.; SIDIKE, P.; NASRIN, M. S.; ESESN, B. C. V.; AWWAL, A. A. S.; ASARI, V. K. The history began from alexnet: A comprehensive survey on deep learning approaches. **arXiv preprint arXiv:1803.01164**, 2018.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **arXiv preprint arXiv:1409.0473**, 2014.

BECKER, S.; HUG, R.; HÜBNER, W.; ARENS, M. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. **arXiv preprint arXiv:1805.07663**, 2018.

BENESTY, J.; CHEN, J.; HUANG, Y.; COHEN, I. Pearson correlation coefficient. *In*: **Noise reduction in speech processing**. [*S.l.: s.n.*]: Springer, 2009. p. 1–4.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1798–1828, 2013.

BHATTACHARYYA, A.; FRITZ, M.; SCHIELE, B. Long-term on-board prediction of people in traffic scenes under uncertainty. *In*: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [*S.l.: s.n.*], 2018. p. 4194–4202.

BHATTACHARYYA, A.; REINO, D. O.; FRITZ, M.; SCHIELE, B. Euro-pvi: Pedestrian vehicle interactions in dense urban centers. *In*: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [*S.l.: s.n.*], 2021. p. 6408–6417.

CAI, Y.; DAI, L.; WANG, H.; CHEN, L.; LI, Y.; SOTELO, M. A.; LI, Z. Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, 2021.

CAO, D.; FU, Y. Using graph convolutional networks skeleton-based pedestrian intention estimation models for trajectory prediction. *In*: IOP PUBLISHING. **Journal of Physics: Conference Series**. [*S.l.: s.n.*], 2020. v. 1621, n. 1, p. 012047.

Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.; Sheikh, Y. A. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2019.

CASTRO., A.; GRASSI, V.; PONTI, M. Deep depth completion of low-cost sensor indoor rgb-d using euclidean distance-based weighted loss and edge-aware refinement. *In*: INSTICC. **Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,**. [*S.l.: s.n.*]: SciTePress, 2022. p. 204–212. ISBN 978-989-758-555-5.

CASTRO, A. R.; GRASSI, V. Learning to estimate multivariate uncertainty in deep pedestrian trajectory prediction. *In*: **2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)**. [*S.l.: s.n.*], 2023. p. 415–420.

CHEN, S.; XIE, E.; GE, C.; CHEN, R.; LIANG, D.; LUO, P. CycleMLP: A MLP-like architecture for dense prediction. *In*: **International Conference on Learning Representations**. [*S.l.: s.n.*], 2022. Disponível em: https://openreview.net/forum?id=NMEceG4v69Y.

CHEN, T.; TIAN, R. A survey on deep-learning methods for pedestrian behavior prediction from the egocentric view. *In*: IEEE. **2021 IEEE International Intelligent Transportation Systems Conference (ITSC)**. [*S.l.: s.n.*], 2021. p. 1898–1905.

CHENG, H.; LIAO, W.; TANG, X.; YANG, M. Y.; SESTER, M.; ROSENHAHN, B. Exploring dynamic context for multi-path trajectory prediction. *In*: IEEE. **2021 IEEE International Conference on Robotics and Automation (ICRA)**. [*S.l.: s.n.*], 2021. p. 12795–12801.

CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S.; SCHIELE, B. The cityscapes dataset for semantic urban scene understanding. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [*S.l.: s.n.*], 2016.

CZECH, P.; BRAUN, M.; KRESSEL, U.; YANG, B. Behavior-aware pedestrian trajectory prediction in ego-centric camera views with spatio-temporal ego-motion estimation. **Machine Learning and Knowledge Extraction**, MDPI, v. 5, n. 3, p. 957–978, 2023.

DONG, S.; WANG, P.; ABBAS, K. A survey on deep learning and its applications. **Computer Science Review**, Elsevier, v. 40, p. 100379, 2021.

GAL, Y.; GHAHRAMANI, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *In*: PMLR. **international conference on machine learning**. [*S.l.: s.n.*], 2016. p. 1050–1059.

GAL, Y.; GHAHRAMANI, Z. A theoretically grounded application of dropout in recurrent neural networks. **Advances in neural information processing systems**, v. 29, 2016.

GERS, F. A.; SCHMIDHUBER, J. Recurrent nets that time and count. *In*: IEEE. **Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium**. [*S.l.: s.n.*], 2000. v. 3, p. 189–194.

GIRIN, L.; LEGLAIVE, S.; BIE, X.; DIARD, J.; HUEBER, T.; ALAMEDA-PINEDA, X. Dynamical variational autoencoders: A comprehensive review. **arXiv preprint arXiv:2008.12595**, 2020.

GIULIARI, F.; HASAN, I.; CRISTANI, M.; GALASSO, F. Transformer networks for trajectory forecasting. *In*: IEEE. **2020 25th International Conference on Pattern Recognition (ICPR)**. [*S.l.: s.n.*], 2021. p. 10335–10342.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [*S.l.: s.n.*]: Editora Blucher, 2000.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.: s.n.*]: MIT Press, 2016. http://www.deeplearningbook.org.

GOUDA, M.; MIRZA, J.; WEISS, J.; CASTRO, A. R.; EL-BASYOUNY, K. Octree-based point cloud simulation to assess the readiness of highway infrastructure for autonomous vehicles. **Computer-Aided Civil and Infrastructure Engineering**, Wiley Online Library, v. 36, n. 7, p. 922–940, 2021.

GULZAR, M.; MUHAMMAD, Y.; MUHAMMAD, N. A survey on motion prediction of pedestrians and vehicles for autonomous driving. **IEEE Access**, IEEE, 2021.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2016. p. 770–778.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HOUDT, G. V.; MOSQUERA, C.; NÁPOLES, G. A review on the long short-term memory model. **Artificial Intelligence Review**, Springer, v. 53, n. 8, p. 5929–5955, 2020.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2017. p. 4700–4708.

ILG, E.; CICEK, O.; GALESSO, S.; KLEIN, A.; MAKANSI, O.; HUTTER, F.; BROX, T. Uncertainty estimates and multi-hypotheses networks for optical flow. *In*: **Proceedings of the European Conference on Computer Vision (ECCV)**. [*S.l.: s.n.*], 2018. p. 652–667.

JEBAMIKYOUS, H.-H.; KASHEF, R. Autonomous vehicles perception (avp) using deep learning: Modeling, assessment, and challenges. **IEEE Access**, IEEE, v. 10, p. 10523–10535, 2022.

KENDALL, A.; GAL, Y. What uncertainties do we need in bayesian deep learning for computer vision? **Advances in neural information processing systems**, v. 30, 2017.

KENDALL, A.; GAL, Y.; CIPOLLA, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2018. p. 7482–7491.

KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. **arXiv preprint arXiv:1312.6114**, 2013.

KINGMA, D. P.; WELLING, M. An introduction to variational autoencoders. **arXiv preprint arXiv:1906.02691**, 2019.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural computation**, MIT Press, v. 1, n. 4, p. 541–551, 1989.

LIAN, D.; YU, Z.; SUN, X.; GAO, S. As-mlp: An axial shifted mlp architecture for vision. **arXiv preprint arXiv:2107.08391**, 2021.

LIN, T.; WANG, Y.; LIU, X.; QIU, X. A survey of transformers. **arXiv preprint arXiv:2106.04554**, 2021.

LIU, H.; DAI, Z.; SO, D.; LE, Q. Pay attention to mlps. **Advances in Neural Information Processing Systems**, v. 34, 2021.

MANGALAM, K.; AN, Y.; GIRASE, H.; MALIK, J. From goals, waypoints & paths to long term human trajectory forecasting. *In*: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [*S.l.: s.n.*], 2021. p. 15233–15242.

MANGALAM, K.; GIRASE, H.; AGARWAL, S.; LEE, K.-H.; ADELI, E.; MALIK, J.; GAIDON, A. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *In*: SPRINGER. **European Conference on Computer Vision**. [*S.l.: s.n.*], 2020. p. 759–776.

MARKATOU, M.; KARLIS, D.; DING, Y. Distance-based statistical inference. **Annual Review of Statistics and its Application**, Annual Reviews, v. 8, p. 301–327, 2021.

MELLO, R. F. de; PONTI, M. A. Statistical learning theory. **Machine Learning**, Springer, 2018.

NIU, Z.; ZHONG, G.; YU, H. A review on the attention mechanism of deep learning. **Neurocomputing**, Elsevier, v. 452, p. 48–62, 2021.

POOLE, D.; MACKWORTH, A.; GOEBEL, R. Computational intelligence. 1998.

RASOULI, A.; KOTSERUBA, I.; KUNIC, T.; TSOTSOS, J. K. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. *In*: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [*S.l.: s.n.*], 2019. p. 6262–6271.

REZENDE, D. J.; MOHAMED, S.; WIERSTRA, D. Stochastic backpropagation and approximate inference in deep generative models. *In*: PMLR. **International conference on machine learning**. [*S.l.: s.n.*], 2014. p. 1278–1286.

RIDEL, D.; DEO, N.; WOLF, D.; TRIVEDI, M. Scene compliant trajectory forecast with agent-centric spatio-temporal grids. **IEEE Robotics and Automation Letters**, IEEE, v. 5, n. 2, p. 2816–2823, 2020.

RIDEL, D.; REHDER, E.; LAUER, M.; STILLER, C.; WOLF, D. A literature review on the prediction of pedestrian behavior in urban scenarios. *In*: IEEE. **2018 21st International Conference on Intelligent Transportation Systems (ITSC)**. [*S.l.: s.n.*], 2018. p. 3105–3112.

RIDEL, D. A.; DEO, N.; WOLF, D.; TRIVEDI, M. Understanding pedestrian-vehicle interactions with vehicle mounted vision: An lstm model and empirical analysis. *In*: IEEE. **2019 IEEE Intelligent Vehicles Symposium (IV)**. [*S.l.: s.n.*], 2019. p. 913–918.

RUDENKO, A.; PALMIERI, L.; HERMAN, M.; KITANI, K. M.; GAVRILA, D. M.; ARRAS, K. O. Human motion trajectory prediction: A survey. **The International Journal of Robotics Research**, Sage Publications Sage UK: London, England, v. 39, n. 8, p. 895–935, 2020.

RUSSELL, R. L.; REALE, C. Multivariate uncertainty in deep learning. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, 2021.

SALZMANN, T.; IVANOVIC, B.; CHAKRAVARTY, P.; PAVONE, M. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *In*: SPRINGER. **European Conference on Computer Vision**. [*S.l.: s.n.*], 2020. p. 683–700.

SANTOS, A. C. D.; GRASSI, V. Pedestrian trajectory prediction with pose representation and latent space variables. *In*: IEEE. **2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)**. [*S.l.: s.n.*], 2021. p. 192–197.

SIGHENCEA, B. I.; STANCIU, R. I.; CĂLEANU, C. D. A review of deep learning-based methods for pedestrian trajectory prediction. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 21, n. 22, p. 7543, 2021.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A.; LIBONI, L. H. B.; ALVES, S. R. Artificial neural networks: a practical course. **Switzerland: Springer International Publishing**, 2017.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SU, Z.; HUANG, G.; ZHANG, S.; HUA, W. Crossmodal transformer based generative framework for pedestrian trajectory prediction. *In*: IEEE. **2022 International Conference on Robotics and Automation (ICRA)**. [*S.l.: s.n.*], 2022. p. 2337–2343.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. **Advances in neural information processing systems**, v. 27, 2014.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2015. p. 1–9.

TOLSTIKHIN, I. O.; HOULSBY, N.; KOLESNIKOV, A.; BEYER, L.; ZHAI, X.; UNTERTHINER, T.; YUNG, J.; STEINER, A.; KEYSERS, D.; USZKOREIT, J. *et al.* Mlp-mixer: An all-mlp architecture for vision. **Advances in Neural Information Processing Systems**, v. 34, 2021.

TRUCCO, E.; VERRI, A. **Introductory techniques for 3-D computer vision**. [*S.l.: s.n.*]: Prentice hall Englewood Cliffs, 1998. v. 201.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WANG, C.; WANG, Y.; XU, M.; CRANDALL, D. Stepwise goal-driven networks for trajectory prediction. **IEEE Robotics and Automation Letters**, IEEE, 2022.

WANG, Z.; PAPANIKOLOPOULOS, N. Estimating pedestrian crossing states based on single 2d body pose. *In*: IEEE. **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [*S.l.: s.n.*], 2020. p. 2205–2210.

YAO, Y.; ATKINS, E.; JOHNSON-ROBERSON, M.; VASUDEVAN, R.; DU, X. Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. **IEEE Robotics and Automation Letters**, IEEE, v. 6, n. 2, p. 1463–1470, 2021.

YAO, Y.; XU, M.; WANG, Y.; CRANDALL, D. J.; ATKINS, E. M. Unsupervised traffic accident detection in first-person videos. *In*: IEEE. **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [*S.l.: s.n.*], 2019. p. 273–280.

YU, T.; LI, X.; CAI, Y.; SUN, M.; LI, P. Rethinking token-mixing mlp for mlp-based vision backbone. **arXiv preprint arXiv:2106.14882**, 2021.

YU, T.; LI, X.; CAI, Y.; SUN, M.; LI, P. S2-mlp: Spatial-shift mlp architecture for vision. *In*: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision**. [*S.l.: s.n.*], 2022. p. 297–306.

YUAN, Y.; WENG, X.; OU, Y.; KITANI, K. M. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. *In*: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [*S.l.: s.n.*], 2021. p. 9813–9823.

ZHAO, R.; YAN, R.; WANG, J.; MAO, K. Learning to monitor machine health with convolutional bi-directional lstm networks. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 17, n. 2, p. 273, 2017.


EESC • USP