Universidade Federal do Paraná

Douglas Zaia Adam e Henrique Colodetti Escanferla

Deep Learning em classificação de imagens com ruído

Curitiba PR
2017

Douglas Zaia Adam e Henrique Colodetti Escanferla

Deep Learning em classificação de imagens com ruído

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Jaques Spinosa.

Curitiba PR
2017

Resumo

Em aprendizado de máquina e reconhecimento de padrões as redes neurais convolucionais mostraram os melhores resultados em tarefas difíceis em visão computacional. Nestas tarefas é importante garantir que as técnicas usadas sejam robustas a ruídos, pois em situações reais eles existem em abundância. A proposta deste trabalho foi realizar um estudo de três modelos distintos quanto sua robustez com ruídos *Gaussian Noise* e *Motion Blur* em imagens de dígitos manuscritos e objetos cotidianos. Utilizando as bases MNIST e CIFAR-10, um modelo de regressão *Softmax*, um modelo *Multilayer Perceptron* e um modelo de *Deep Learning*, demonstramos como estes reagem aos ruídos nas imagens.

Palavras-chave: Rede Neural, Deep Learning, MNIST, CIFAR-10, Imagens com ruídos.

Abstract

In machine learning and pattern recognition the convolutional neural networks showed the best results in difficult tasks in computer vision. In these tasks it's important to ensure that the techniques used are robust to noises, because in real situations they exist in abundance. The purpose of this experiment was to perform a study of two distinct models regarding their robustness with common noises in images of handwritten digits and daily objects. Using the MNIST database and the CIFAR-10 database, a simple softmax regression model and a deep learning model, we demonstrate how these react to the noises in the images.

Keywords: Neural Network, Deep Learning, MNIST, CIFAR-10, Images with noises.

Sumário

Lista de Figuras

Lista de Acrônimos

DINF	Departamento de Informática
PPGINF	Programa de Pós-Graduação em Informática
UFPR	Universidade Federal do Paraná

Lista de Símbolos

σ	desvio padrão
μ	média

Capítulo 1

Introdução

Na área de visão computacional um dos maiores desafios é o reconhecimento de padrões. Os modelos de redes neurais, principalmente os de *deep learning*, trouxeram resultados expressivos e são cada vez mais estudados e usados em pesquisas cientificas e aplicações práticas, como: classificação de objetos em imagens, transcrição de voz para texto, cruzamento de tópicos, postagens e produtos com os gostos de usuários, tornar as buscas mais relevantes. (14)

Nos problemas do mundo real, boas propostas de reconhecimento de objetos frequentemente demonstram limitações quando apresentados dados fora do escopo, como por exemplo imagens com planos de fundo não vistos pelos modelos junto com ruídos além de proporções dos objetos diferentes da base de imagens utilizada. Para a utilização destas técnicas na prática, os métodos de reconhecimento de objetos devem ser capazes de lidar com situações inesperadas. (7)

Com isto, este estudo tem como objetivo realizar comparações entre modelos de reconhecimento de caractéres com ruídos em diferentes intensidades utilizando as conhecidas bases MNIST e CIFAR-10 e o framework de aprendizado de máquina da Google, o Tensorflow. (LeCun et al.) (Google)

Este trabalho é dividido em capítulos. O capítulo dois discorre sobre as redes neurais e o modelo *deep learning* para depois contextualizar o reconhecimento de caracteres e discutir a base MNIST, a base CIFAR-10 e os ruídos mais comuns em imagens. Por fim, apresentamos os modelos utilizados e os experimentos com seus resultados de desempenho conforme alteramos os parâmetros dos ruídos gerados nas bases.

Capítulo 2

Fundamentação Teórica

2.1 Redes Neurais Artificiais

Uma rede neural artificial consiste num modelo de aprendizado bioinspirado composto por um grupo de módulos processadores de informação chamados de neurônios artificiais que se conectam formando uma rede. Uma versão simples de um neurônio artificial possui várias entradas (x1, x2, x3, ..., xp) com seus respectivos pesos (w1, w2, w3, ..., wp), um valor independente w0 chamado de bias e uma saída y. (3)

Uma visão básica de um neurônio artificial, nesta forma simples chamada de perceptron, é mostrada na seguinte figura:

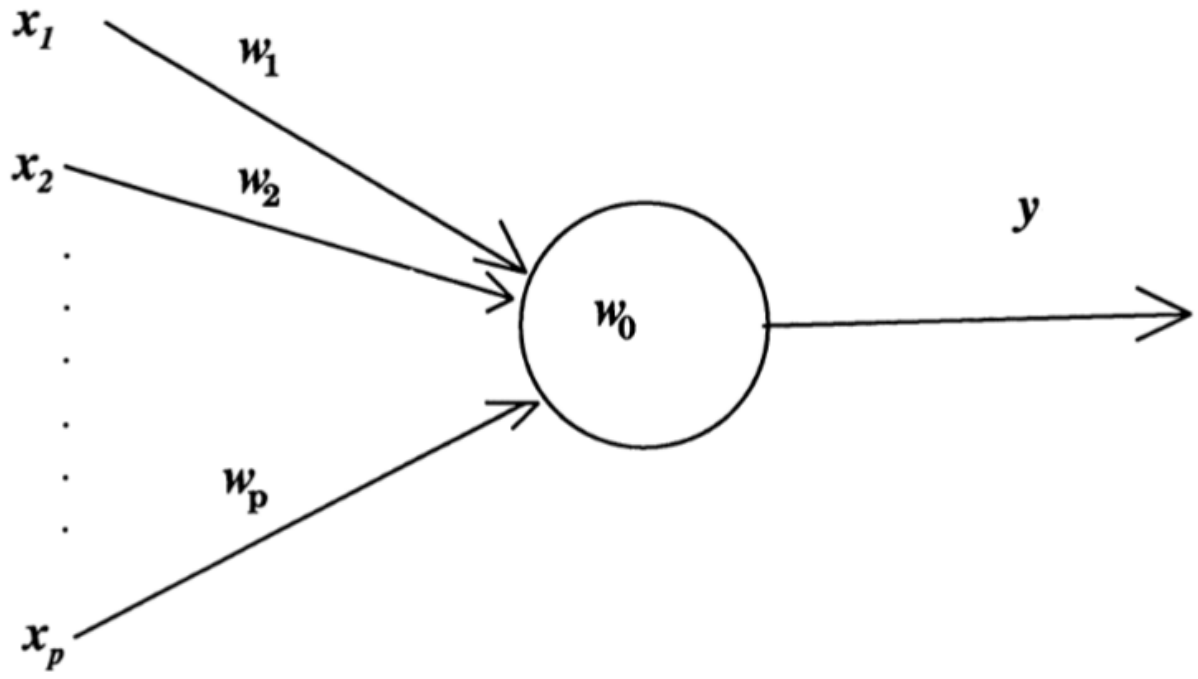

Figura 2.1: Esquema básico de um perceptron. (3)

O valor da saída, y, resulta da soma dos produtos das entradas pelos seus pesos somados ao bias w0. É comum o uso de uma função de ativação na saída dos neurônios artificiais que recebem y e produzem um valor final z de acordo com uma forma de gráfico específica. A função de ativação mais usada em redes neurais artificiais recentemente é a função retificadora que é

similar a função retificadora de meia onda. Nesta função, se y é menor que 0 então z é igual a 0. Caso contrário, z recebe o valor de y. (14)

Uma rede neural artificial é dividida em grupos de neurônios artificiais chamados de camadas neurais como mostra a figura abaixo de uma rede neural artificial simples do tipo feedforward:

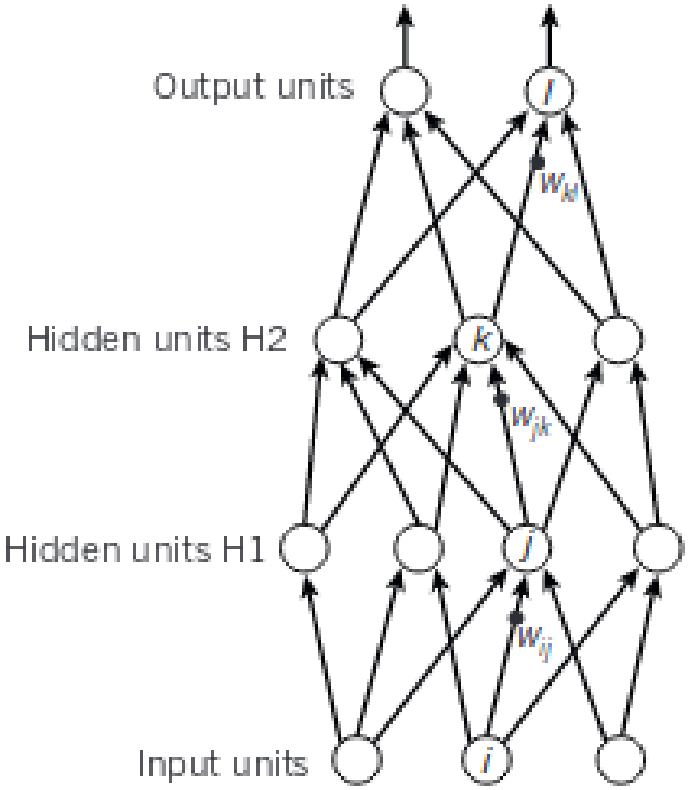

Figura 2.2: Ilustração de uma rede neural artificial feedforward. (14)

Estas camadas neurais são divididas em 3 tipos:

- de entrada, que recebem sinais do ambiente e enviam as saídas para as camadas escondidas.
- escondida, que recebem sinais da camada de entrada ou de outra camada escondida e enviam as saídas para outra camada escondida ou para a camada de saída.
- de saída, que recebem sinais da camada escondida e produzem as saídas que compõem a resposta da rede ao estímulo recebido do ambiente. (3) (14)

Como visto na figura, uma rede neural artificial possui uma camada de entrada e outra de saída conectadas por várias camadas intermediárias escondidas.

2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais demonstram bom desempenho na extração de características, na detecção de representações com multiplos níveis de abstração e na classificação

dos dados, isto é, na detecção de padrões num conjunto de dados com várias dimensões, como é o caso de sinais, sons, imagens e vídeos, seguido da classificação deles.

Os aspectos principais que tornam este tipo de rede neural artificial um bom modelo de detecção de padrões são as conexões locais, os pesos compartilhados das conexões neurais, a técnica de pooling e o empilhamento de várias camadas neurais que provocam o efeito de convolução da informação conforme ela percorre a rede neural artificial convolucional.

A estrutura básica de uma rede neural articial convolucional é formada por uma sequência de estágios, como apresentado na Figura 2.3. A entrada é uma imagem de um cachorro, cuja a raça é parecida com um lobo selvagem, que possui três canais do padrão RGB. Estes canais de dados são inseridos na primeira camada neural da rede que sofrem diversas transformações conforme percorrem a rede até o final onde obtemos pontuações para as classes disponíveis que categorizam a imagem.

Os primeiros estágios são compostos por dois tipos de camadas neurais: as camadas convolucionais e as camadas de pooling.

As camadas convolucionais, de uma forma simples, consistem num conjunto mapeado de características retiradas dos dados recebidos. Cada mapa possui uma matriz filtro chamada de kernel composta por pesos que, executando uma multiplicação matricial com os dados, resultam num conjunto de características. Após a camada convolucional, os dados passam pela função de ativação retificadora e segue para a camada de pooling.

Na camada de pooling usa-se uma certa estratégia para simplificar os dados e extrair partes menores deles sem perder as características detectadas da camada anterior. Uma consequência direta deste processo é a redução das dimensões dos dados como no caso da figura que mostra a imagem de entrada diminuindo de tamanho conforme atravessa a rede neural artificial convolucional. A estratégia geralmente utilizada é a maximização que, a cada parte seccionada dos dados recebidos, seleciona o maior valor para compor os dados de saída que são repassados para a próxima camada neural. Tal processo mantém as características detectadas mesmo que ocorram certas variações nos dados de entrada e reduz a quantidade de dados passados para as camadas neurais seguintes.

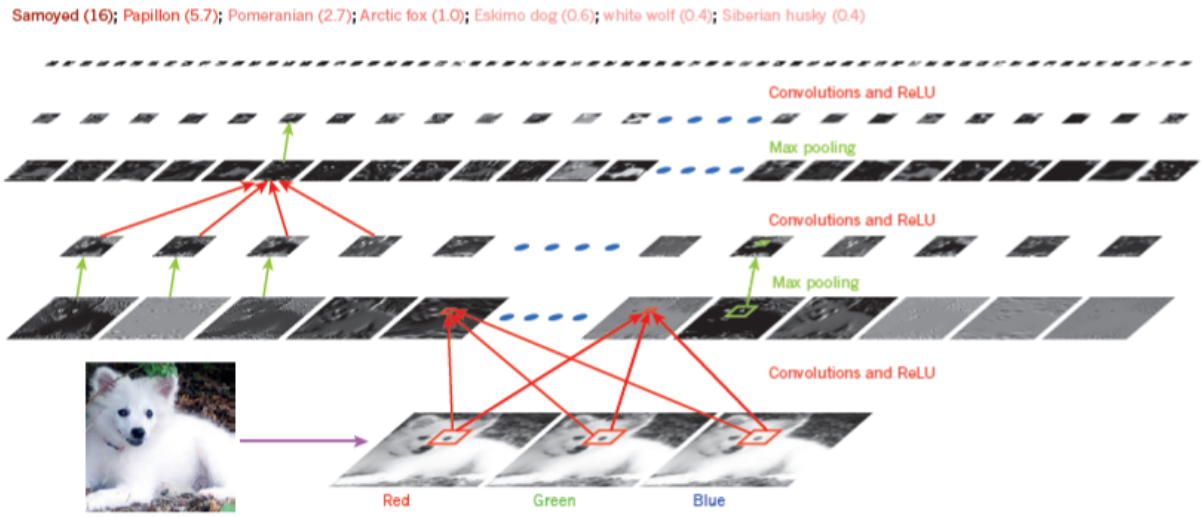

Figura 2.3: Esquema do processo de convolução de uma imagem. (14)

Com algumas camadas neurais empilhadas de convolução, retificação e pooling junto com camadas neurais completamente conectadas forma-se um modelo padrão de uma rede neural artificial convolucional. O número de saídas da última camada neural completamente

conectada corresponde a um valor númerico que a rede produz para classificar a imagem de entrada como no exemplo da figura. Mesmo a imagem sendo de um cachorro da raça *Samoyed* parecida com um lobo selvagem, a rede consegue detectar e diferenciar suas características de outras raças de cães parecidos com lobos selvagens, como o *Husky* siberiano. Essa habilidade de classificação se mantém quando alteramos o fundo da imagem onde se encontra o animal ou o angulo da fotografia. Tal técnica é uma das mais conhecidas e categorizadas como *deep learning*. De poucas décadas para a atualidade é a técnica que mais se destaca pelo desempenho obtido em reconhecimento de padrões e extração de características de conjuntos grandes de dados multidimensionais. (14)

2.3 Inspiração Biológica

Observando a anatomia de um neurônio humano, podemos observar sua semelhança com a estrutura básica de um perceptron.

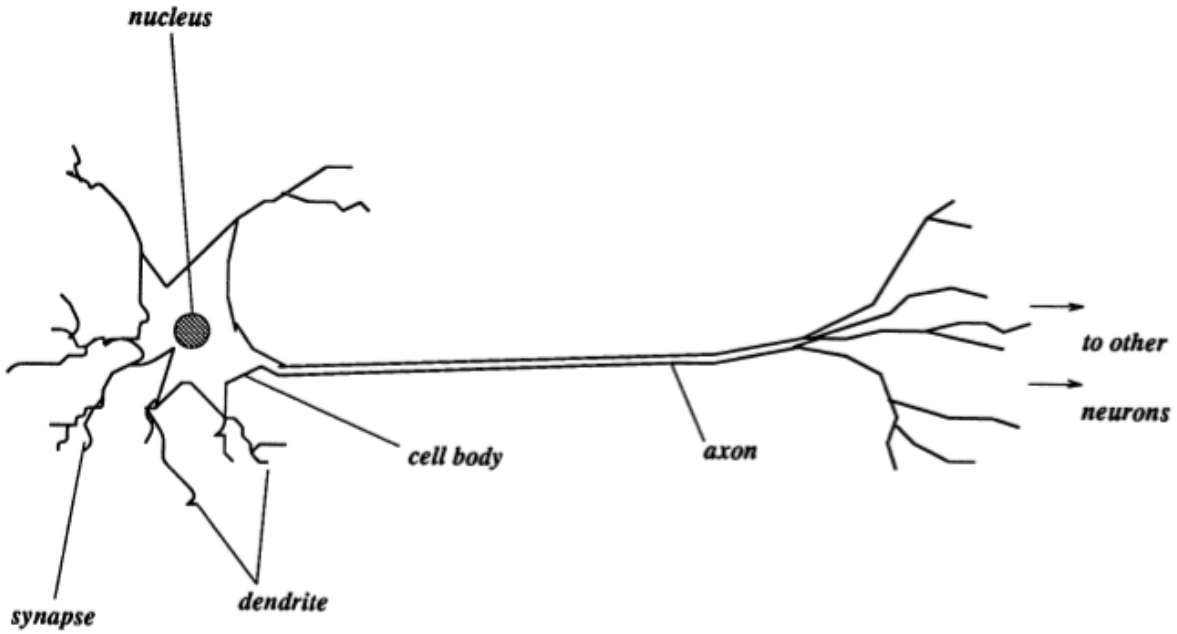

Figura 2.4: Ilustração anatômica de um neurônio humano. (3)

Como no perceptron, um neurônio humano possui várias conexões de entrada e uma única saída. Isso não impede que esta saída seja transmitida para mais de um neurônio como ocorre em seres vivos e em redes neurais artificiais.

A concepção de neurônios artificiais e conexões neurais formando uma rede neural artificial como forma de processar e inferir informações vem de estudos da neurociência que inspiraram questionamentos na área de inteligência artificial sobre as diferenças de desempenho entre um cérebro humano e uma máquina de Von Neumann, modelo teórico de um computador.

Por exemplo, para contas matemáticas simples, qualquer microchip moderno consegue processar e mostrar o resultado em nanossegundos enquanto que um cérebro humano demora cerca de alguns milissegundos para esta mesma tarefa. Por outro lado, em tarefas cognitivas e processamento de inferências inteligentes, os computadores sofrem em resolver enquanto que um ser humano facilmente encontra uma boa resposta.

O ponto importante que justifica esta diferença de desempenho está, não na velocidade de processamento das informações recebidas, mas na forma como ela é organizada e processada.

Esta ideia vem da noção do paralelismo e do conexionismo que uma rede neural possui. Todo o trabalho é dividido entre um bom número de neurônios e cada um deles possui várias conexões com outros neurônios. Qualquer neurônio pode ser estimulado e estimular qualquer outro neurônio. Isto propicia um poder de processamento grande por toda a rede neural.

Além disso, outras características importantes da atividade neural são as habilidades de adaptação e auto organização. De acordo com experiências passadas, a estrutura da rede se adapta para obter melhores resultados nas mesmas tarefas. (3)

As camadas neurais convolucionais e de pooling são inspiradas na idéia de células simples e complexas da neurosciência visual e a estrutura geral é baseada em hierarquias estruturais vistas no cortex visual. É visto que no cortex visual de gatos, existem dois grupos de células neurais com funções distintas que apresentam conexões entre si. Um grupo é caracterizado como simples e o outro como complexo. (9) Isto mostra semelhança com as diferentes camadas neurais em uma rede neural convolucional que dividem funções diferentes: extrair características e classificar os dados.

Por fim, mostrar as mesmas imagens para redes neurais artificiais convolucionais e macacos apontam que as ativações neurais são, até certo ponto, semelhantes. Isto nunca foi alcançado por nenhum outro modelo de reconhecimento de padrões bioinspirado. (2)

Capítulo 3

Trabalhos Relacionados

Em (6) apresentam-se taxas de acerto de modelos proeminentes, como a Google Net, no reconhecimento de padrões utilizando a base MNIST. Usando Rede Neural Convolucional, com o Tensorflow, a taxa foi de 98,7%, usando Máquina de Vetor de Suporte a taxa foi de 97,91%, usando K Vizinhos mais Próximos a taxa foi de 96,61% e usando o Classificador de Florestas Aleatórias a taxa foi de 96,89%.

Em (1) foi usado um modelo de Rede Neural Convolucional na base de imagens notMNIST que consiste de um conjunto de imagens mais complexas utilizada como ferramenta para avaliar a performance de modelos de reconhecimento de padrões de uma forma menos convencional se comparado a comum tarefa de comparar performances de modelos no MNIST. A taxa foi de 98%.

Em (5), foram usados quatro modelos conhecidos de redes neurais convolucionais próximos ao estado da arte com seus pesos já treinados com a base de dados *ImageNet* sem ruídos. Estes modelos treinados estão disponíveis na biblioteca de aprendizado de máquina Caffe. Na medição das taxas de performance foi usado um subconjunto de imagens do conjunto total de testes da base de dados além de selecionar somente alguns exemplos de cada classe deste subconjunto com o objetivo de poupar tempo de execução. (4) (11)

Mostrou-se a variação das taxas de performance em 5 tipos de ruído em vários níveis. Foram usados os ruídos de *Gaussian Blur*, *Gaussian noise*, redução de contraste, compressão JPEG e compressão JPEG2000. Nos ruídos gaussianos somente o desvio padrão foi ajustado.

Os resultados mostram que os ruídos de *Gaussian Blur* e *Gaussian noise* causam mais impacto nos modelos do que os ruídos da redução de contraste, compressão JPEG e compressão JPEG2000 e isto é um indicativo da viabilidade em comprimir grandes bases de dados sem perder muito do desempenho dos modelos.

Capítulo 4

Ruídos e Bases

4.1 Bases

Neste trabalho foram utilizadas duas bases de imagens conhecidas no ambiente acadêmico para realizar os experimentos: MNIST e CIFAR-10.

4.1.1 MNIST

MNIST (*Modified National Institute of Standards and Technology database*) é uma base numérica manuscrita onde cada classe representa um digito entre 0 à 9. Cada imagem possui um único digito, como consequência uma única classe, e tamanho igual à 28x28 pixels. Geralmente é separada em duas partes, uma para treino com 60 mil exemplos e outra para teste com 10 mil exemplos. A base já vem dividida de modo que não aconteça do treino possuir muitos exemplos de uma classe e no teste poucos da mesma e vice-versa.

Figura 4.1: Amostra de todas as classes da base MNIST.

4.1.2 CIFAR 10

CIFAR-10 (*Canadian Institute For Advanced Research*) é uma base de imagens com 10 classes: avião, automóvel, passaro, gato, cervo, cachorro, sapo, cavalo, barco e caminhão. Com tamanho de 32x32 pixels, as imagens seguem o padrão RGB, possuindo três canais de cores: vermelho, verde e azul. O treino possui exatamente 5 mil exemplos de cada classe e é separado em 5 pacotes com 10 mil exemplos cada totalizando 50 mil, e o teste contém mil exemplos de cada classe totalizando 10 mil exemplos. (13)

Figura 4.2: Amostra de todas as classes da base CIFAR 10.

4.2 Ruídos

Em uma visão geral, o ruído é a perda de informação que um dado apresenta. Este dado pode ser uma foto, um documento de texto, uma música etc. Sendo mais específico para imagens, os ruídos danificam a integridade do conteúdo, assim pioram a qualidade do que está sendo representado. A perda da informação varia, pois existem vários tipos de ruído e intensidade, o que acarreta em um problema, se a perda de conteúdo for grande pode invalidar a imagem.

Existem vários motivos para uma imagem apresentar ruídos, seja pela falta de qualidade de uma câmera, conversões de imagens para formatos mais econômicos de memória, questões naturais do ambiente no momento em que a imagem foi capturada, baixa resolução, entre outros.

Este trabalho utilizou dois ruídos conhecidos para realizar os experimentos, o *Gaussian Noise* e o *Motion Blur*.

4.2.1 Motion Blur

O *Motion Blur* é causado pelo movimento relativo entre a câmera e o objeto retratado durante o tempo em que a imagem está sendo capturada. (10)

As figuras 4.3 e 4.4 abaixo mostram diferentes níveis deste ruído em uma amostra de cada classe da base MNIST e CIFAR-10:

Figura 4.3: Diferentes níveis do *Motion Blur* na base MNIST.

Figura 4.4: Diferentes níveis do *Motion Blur* na base CIFAR-10.

Este ruído causa uma sensação de movimento para a imagem, fazendo com que ela fique borrada e quanto maior a intesidade mais difícil de perceber os elementos contidos nela. Para o trabalho foi escolhida a orientação à direita, então todas as imagens parecem estar se deslocando neste sentido e a cada intensidade era deslocado 6 pixels, começando de 0 até 24.

4.2.2 Gaussian Noise

O *Gaussian Noise* é um ruído que utilizada a função densidade de probabilidade da distribuição normal e é descrita pela equação (4.1).

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{4.1}$$

As figuras 4.5 e 4.6 abaixo mostram diferentes níveis deste ruído em uma amostra de cada classe da base MNIST e CIFAR-10:

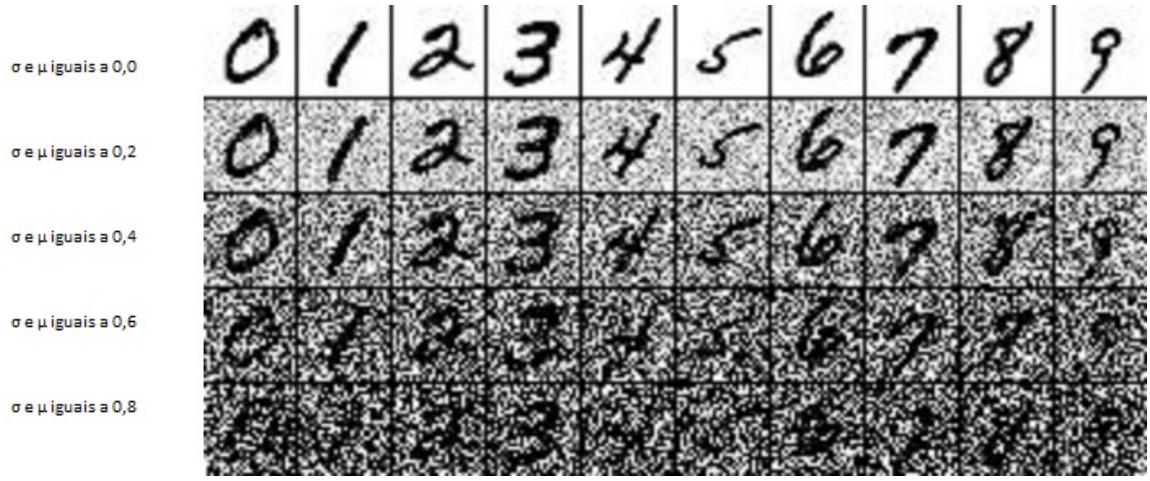

Figura 4.5: Diferentes níveis do *Gaussian Noise* na base MNIST.

Figura 4.6: Diferentes níveis do *Gaussian Noise* na base CIFAR-10.

Utilizando a equação (4.1) faz com que a imagem ganhe vários pontos em tons de cinza para o MNIST e coloridos para o CIFAR-10. Isto ocorre devido ao formato de cor que cada base apresenta. Para gerar os 5 níveis de *Gaussian Noise* apresentados foram modificados os parâmetros média e desvio padrão, ambos começando de 0 até 0,8.

Com grande intesidade o *Gaussian Noise* faz com que a legibilidade das imagens seja muito ruim, como por exemplo: o cavalo que só consegue ver o contorno dele nos últimos níveis.

Capítulo 5

Experimentos

5.1 Aspectos gerais

Os códigos dos modelos foram estruturados para receberem as imagens da base MNIST como entrada de dados e foram feitas alterações nos modelos para utilizarem as imagens da base CIFAR-10 pois o tamanho dos exemplos diferem de 28 por 28 pixels para 32 por 32 pixels.

Adicionalmente, foram modificados para lerem os dados das imagens em CSV convertendo-os para o formato esperado pelos modelos. Além disso, foram inseridas estruturas de repetição para executar as iterações e épocas do processo de treinamento armazenando os desempenhos de cada iteração. Os testes foram feitos tirando as médias e o desvios padrão das taxas de desempenho dos modelos. A divisão dos dados em um conjunto de treinamento e um conjunto de teste seguiu a divisão original já presente nas bases.

O número de iterações executadas a cada etapa da coleta das métricas de desempenho foi cortado de 10, um valor geralmente usado, para 5 com o intuito de diminuir o tempo de processamento dos experimentos.

5.2 Modelo Softmax

Foi utilizado o código de um modelo de uma única camada *Softmax* disponível na página web do Tensorflow. (Google)

A função de custo usada foi a entropia cruzada e a técnica de otimização dos pesos foi o gradiente descendente. Ambas as escolhas são comumente usadas em modelos de reconhecimento de padrões.

Os parâmetros usados tomaram os valores padrão do modelo como base. O tamanho do batch foi de 100 imagens. A taxa de aprendizado do otimizador foi de 0.5. O número de épocas foi de 2.

5.3 Modelo *Multilayer Perceptron*

Foi utilizado um modelo comum do tipo *Multilayer Perceptron* com duas camadas escondidas.

A função de custo foi a entropia cruzada como no modelo anterior, mas a técnica de otimização dos pesos foi o *Adam Optimizer* que, apesar da sua semelhança com o gradiente descendente, demonstra uma melhora observável nos desempenhos dos modelos de reconhecimento usando uma taxa de aprendizado inicial que é incrementado exponencialmente ao longo

das fases do treinamento do modelo. Ambas as escolhas são comumente usadas em modelos de reconhecimento de padrões. (12)

Os parâmetros usados tomaram os valores padrão do modelo como base. O tamanho do batch foi de 100 imagens. A taxa de aprendizado do otimizador foi de 0.001. O número de épocas foi de 15.

5.4 Modelo *Deep Learning* com Rede Neural Convolucional

Foi utilizado o código do modelo *deep learning* do Tensorflow. (Google)

A função de custo e a técnica de otimização dos pesos foram, respectivamente, a entropia cruzada e o *Adam Optimizer* como no modelo anterior.

A topologia da rede neural convolucional é, de certa forma, semelhante ao observado nos modelos do LeCun. (14)

A primeira camada do modelo é convolucional computando 32 features com tamanho do patch de 5 por 5 e aplicando a função ReLU no resultado da saída. Logo após há uma camada de max pooling com blocos de tamanho 2 por 2.

A segunda camada do modelo é convolucional computando 64 features com tamanho do patch de 5 por 5 e aplicando, como na camada anterior, a função ReLU no resultado da saída. Novamente seguido de uma camada de max pooling com blocos de tamanho 2 por 2.

Depois temos uma camada neural completamente conectada contendo 1024 neurônios artificiais aplicando a função ReLU na saída. Há uma camada de dropout dos neurônios logo após que sorteia alguns neurônios a ignorarem suas saídas seguindo um parâmetro de probabilidade ajustável. Esta técnica tem o objetivo de evitar o overfitting no modelo embora demonstre uma eficiência considerável em redes neurais grandes o que não é o caso deste modelo usado no experimento.

Por fim, temos uma camada Softmax que reduz o resultado final para 10 valores. Cada um deles, se convertido em porcentagem, representa a chance da imagem pertencer a uma das 10 classes de acordo com a interpretação do modelo.

Os parâmetros usados seguiram os valores padrão do Tensorflow. O tamanho do batch foi de 50 imagens. A taxa de aprendizado inicial do otimizador foi de 0.0001. O número de épocas foi de 20. Por último, a chance do *dropout* ocorrer foi de 50 por cento.

Capítulo 6

Resultados

6.1 Reconhecimento de dígitos no MNIST

Nesta seção, estão expostos os gráficos mostrando as médias das taxas de desempenho e seus respectivos desvios padrão para cada um dos cinco níveis de rúido, cada um dos dois tipos de ruído e cada um dos dois modelos utilizados nos experimentos na base MNIST.

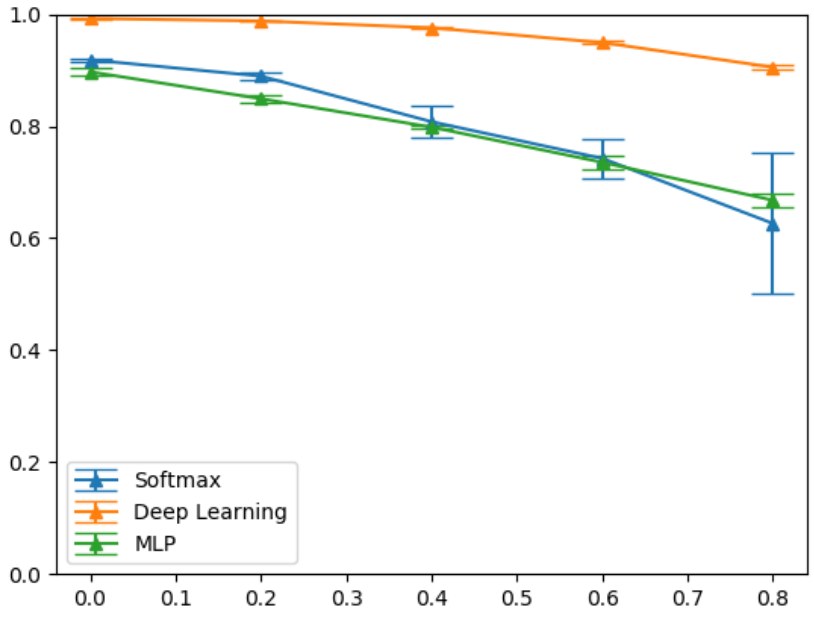

Figura 6.1: Médias de desempenho e desvios padrão na base MNIST com ruído *Gaussian Noise*.

Observa-se que o modelo de *Deep Learning* se sobressai não só pela alta taxa de acerto, mas também pela resistência aos ruídos usados. O ruído *Motion Blur* causa menos impacto nos três modelos.

Considerando os resultados com a base sem ruído e com o máximo de ruído, no modelo *Deep Learning* houve uma variação na média do desempenho de aproximadamente 8.69%, como visto na figura 6.1, e 0.39%, como visto na figura 6.2. No modelo *Multilayer Perceptron* as variações da média do desempenho foram de aproximadamente 22.94%, como visto na figura 6.1, e de 1.86%, como visto na figura 6.2. No modelo *Softmax* as variações da média do desempenho

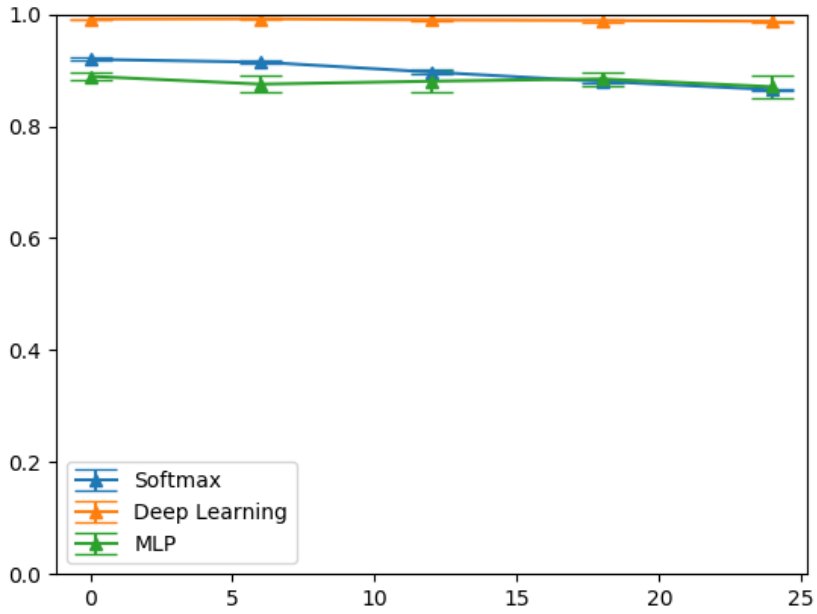

Figura 6.2: Médias de desempenho e desvios padrão na base MNIST com ruído *Motion Blur*.

foram de aproximadamente 29.23%, como visto na figura 6.1, e de 5.42%, como visto na figura 6.2.

Analizando os desvios padrão podemos ver que o modelo *Deep Learning* apresenta maior estabilidade, menores desvios padrão, em suas taxas de desempenho tomando como base as duas figuras desta seção.

Considerando os resultados com a base sem ruído e com o máximo de ruído, para o modelo *Deep Learning*, os desvios tiveram uma diferença de aproximadamente 0.26%, como mostra a figura 6.1, e de 0.08%, como mostra a figura 6.2. No modelo *Multilayer Perceptron* a diferença dos desvios foram de aproximadamente 0.69%, como visto na figura 6.1, e de 1.27%, como visto na figura 6.2. Já para o modelo *Softmax* a diferença dos desvios foram de aproximadamente 12.31%, como visto na figura 6.1, e de 0.05%, como visto na figura 6.2.

Curiosamente, houveram diminuições nos desvios padrão para o *Deep Learning* e *Softmax* destacando maior estabilidade na classificação mesmo com a diminuição da taxa de acerto.

6.2 Reconhecimento de objetos e animais no CIFAR-10

Nesta seção, como na anterior, estão expostos os gráficos mostrando as médias das taxas de desempenho e seus respectivos desvios padrão para cada um dos cinco níveis de rúido, cada um dos dois tipos de ruído e cada um dos dois modelos utilizados nos experimentos com a base CIFAR-10.

Pode-se observar que o modelo de *Deep Learning*, como na seção anterior, é mais eficiente que o modelo *Softmax* e o modelo *Multilayer Perceptron*. O ruído *Motion Blur* causa menos impacto nos três modelos, entretanto isto se apresenta em grau menor que a variação vista nos experimentos com a base MNIST.

Considerando os resultados com a base sem ruído e com o máximo de ruído, no modelo *Deep Learning*, houve uma variação na média do desempenho de aproximadamente 28.66%, como visto na figura 6.3, e 5.73%, como visto na figura 6.4. No modelo *Multilayer Perceptron* as variações da média do desempenho foram de aproximadamente 10.75%, como visto na figura 6.3, e de 1.22%, como visto na figura 6.4. No modelo *Softmax* as variações da média do desempenho foram de aproximadamente 4.21%, como visto na figura 6.3, e de 0.16%, como visto na figura 6.4.

Percebe-se que o modelo *Softmax* apresenta uma taxa de acerto consideravelmente melhor do que a estratégia de classificação aleatória, que seria de 10%, mas é visivelmente pior do que o modelo *Deep Learning*.

Diferentemente do esperado, na média do desempenho do modelo *Softmax*, nota-se um aumento quando é adicionado o primeiro nível de ruído para ambos os ruídos.

Curiosamente a taxa de acerto no *Multilayer Perceptron* é melhor com o primeiro nível de ruído na base em relação a base sem ruído. Também pode-se perceber que o *Multilayer Perceptron* está muito próximo nas taxas de acerto do *Softmax*, e pior que o *Deep Learning*.

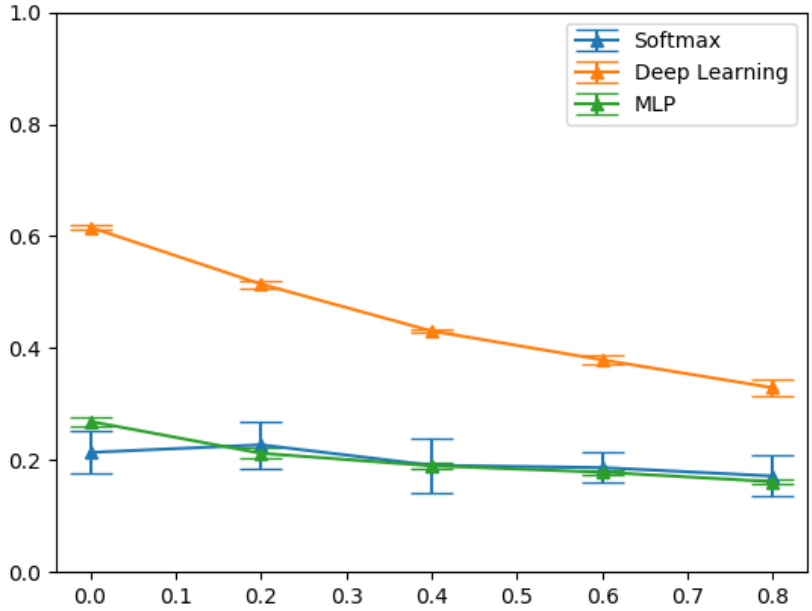

Figura 6.3: Médias de desempenho e desvios padrão na base CIFAR-10 com ruído *Gaussian Noise*.

Analizando os desvios padrão podemos ver que o modelo *Deep Learning* apresenta maior estabilidade, menores desvios padrão, em suas taxas de desempenho tomando como base as duas figuras desta seção.

Mais uma vez, considerando os resultados com a base sem ruído e com o máximo de ruído, para o modelo *Deep Learning*, os desvios tiveram uma diferença de aproximadamente 1.26%, como mostra a figura 6.3, e de 0.28%, como mostra a figura 6.4. Já no modelo *Multilayer Perceptron* a diferença dos desvios foram de aproximadamente 0.54%, como visto na figura 6.3, e de 2.08%, como visto na figura 6.4. E por fim, para o modelo *Softmax* a diferença dos desvios foram de aproximadamente 0.15%, como visto na figura 6.3, e de 0.43%, como visto na figura 6.4.

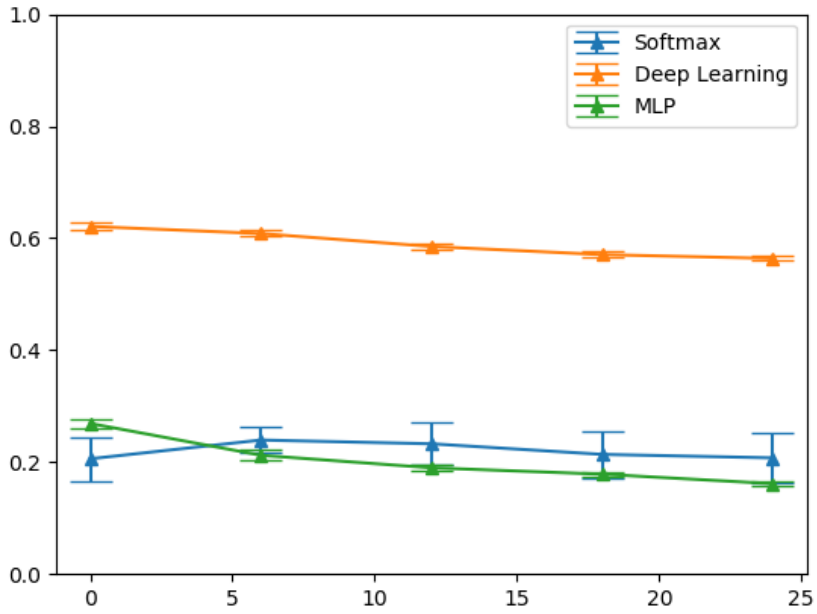

Figura 6.4: Médias de desempenho e desvios padrão na base CIFAR-10 com ruído *Motion Blur*.

O fato diferente do esperado da seção anterior se repete nas diminuições dos desvios padrão de ambos os modelos denotando maior estabilidade na classificação dos dados mesmo com a diminuição da taxa de acerto.

6.3 Considerações Finais

O funcionamento do modelo *Deep Learning* consiste em executar sequencialmente várias abstrações, acumulando alto nível de abstração, passando por vários filtros que tornam possível a detecção de características relevantes para a classificação dos dados de entrada. O funcionamento do modelo *Softmax* se baseia em atribuir pesos que, multiplicados pelo valor bruto de cada *pixel*, representam uma probabilidade de que a imagem seja de tal classe. O ruído *Gaussian Noise* age atribuindo valores aleatórios *pixel* a *pixel* e o ruído *Motion Blur* suaviza as variações entre os *pixels* em uma mesma direção. Estes resultados são justificáveis visto que o primeiro ruído causa muito mais impacto que o outro e o primeiro modelo, em relação ao outro modelo mais simples, tem muito mais capacidade em focar nas caraterísticas que importam e filtrar ruídos que devem ser ignorados para uma boa classificação dos dados.

Comparando as duas bases usadas, vemos que os modelos apresentaram menores desempenhos em acertar as imagens do CIFAR-10 e o modelo *Deep Learning* se sobressai bastante. Isto já é esperado, pois as imagens coloridas do CIFAR-10 apresentam uma quantidade maior de informação para processar e generalizar e também apresentam formas de objetos e seres vivos mais complexos que números manuscritos das imagens do MNIST.

Adicionalmente, deve-se salientar que os efeitos dos ruídos danificaram as imagens do CIFAR-10 de forma consideravelmente maior do que as imagens do MNIST. A inserção do ruído *Gaussian Noise* na base CIFAR-10 apoia este argumento pois foi feito usando os três canais das imagens RGB apresentando danos coloridos e nas imagens monocromáticas da base MNIST as alterações foram feitas na escala de cinza apresentando, então, danos de menor impacto.

Capítulo 7

Conclusão

Com o estudo apresentado neste trabalho pode-se concluir que o *deep learning* é superior aos outros dois modelos, tanto na performance média quanto na estabilidade das performances, devido a sua estrutura de aprendizado apresentando várias camadas com funções diferentes, enquanto o *softmax* só utiliza uma aplicação de pesos nos *pixels* criando um valor de atribuição da imagem para cada classe. O modelo *Multilayer Perceptron* se mostra levemente melhor que o modelo anterior mas ainda é superado com boa margem pelo primeiro modelo. Concluiu-se também que o *Deep Learning* é muito resistente ao ruído *Motion Blur* por ter uma taxa de acerto e desvio padrão muito parecida para todos os níveis deste ruído, mesmo em situações nas quais seria muito difícil para uma pessoa identificar a classe que a imagem representava.

Comparando as bases MNIST e CIFAR-10, é visto que a primeira possui complexidade mais simples como indica as performances mais altas de todos os modelos comparados. No CIFAR-10, além da maior quantidade de dados, o ruído *Gaussian Noise* utilizado causou muito mais impacto do que na base MNIST afinal foi feita separadamente nos três canais de cores das imagens.

Para futuros trabalhos seria interessante utilizar outros ruídos como: *Periodic Noise*, *Anisotropic Noise*, *Purple Fringing*, *Pixelation Noise*, *Chromatic Aberration*, *Sharpening halos*, *Posterization*, *Barrel Distortion*, rotacionamento do conteúdo dentro da imagem, adicionar pequenas imagens ou formas geométricas tentando esconder informações além de utilizar mais de um ruído na mesma imagem para analisar como que os algoritmos se comportam. A adição de mais algoritmos de reconhecimento, como o SVM (*Support Vector Machines*) e o MLP (*Multilayer Perception*) também seria de grande importância para realizar uma comparação mais detalhada sobre quem é mais robusto a ruídos, pois apresentam abordagens, estruturas e técnicas diferentes das utilizadas neste trabalho.

Fica em aberto usar este mesmo ruído de forma uniforme entre os canais ou converter as imagens para escalas de cinza e comparar novamente as performances dos modelos.

Referências Bibliográficas

[1] Bui, V. e Chang, L.-C. (2016). Deep learning architectures for hard character classification. *Int'l Conf. Artificial Intelligence*.

[2] Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J. e DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10:e1003963.

[3] Cheng, B. e Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Statist. Sci.*, 9(1):2–30.

[4] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. e Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Em *CVPR09*.

[5] Dodge, S. F. e Karam, L. J. (2016). Understanding how image quality affects deep neural networks. *CoRR*, abs/1604.04004.

[6] Dutt, A. e AashiDutt (2017). Handwritten digit recognition using deep learning. *International Journal of Advanced Research in Computer Engineering Technology*, 6.

[7] Ghifary, M., Kleijn, W. B. e Zhang, M. (2014). Deep hybrid networks with good out-of-sample object recognition. Em *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 5437–5441.

[Google] Google. Site do projeto livre tensorflow. `https://www.tensorflow.org/`. Acessado em 20/09/2017.

[9] Hubel, D. e Wiesel, T. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154.

[10] Ji, H. e Liu, C. (2008). Motion blur identification from image gradients. Em *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, páginas 1–8. IEEE.

[11] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. e Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. Em *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, páginas 675–678, New York, NY, USA. ACM.

[12] Kingma, D. P. e Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[13] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Relatório técnico.

[14] LeCun, Y., Bengio, Y. e Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444. Insight.

[LeCun et al.] LeCun, Y., Cortes, C. e Burges, C. J. Site do dataset mnist na página do yann lecun. `http://yann.lecun.com/exdb/mnist/`. Acessado em 20/09/2017.