MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENADORIA DE INICIAÇÃO CIENTÍFICA E INTEGRAÇÃO ACADÊMICA
PROGRAMA DE INICIAÇÃO CIENTÍFICA E EM DESENVOLVIMENTO TECNOLÓGICO E INOVAÇÃO

PETERSON WAGNER KAVA DE CARVALHO

RELATÓRIO FINAL

INICIAÇÃO CIENTÍFICA:
PIBIC Voluntária

(08/2017 a 07/2018)

DEEP LEARNING APLICADO A SISTEMAS

Relatório apresentado à Coordenadoria de Iniciação Científica e Integração Acadêmica da Universidade Federal do Paraná por ocasião da conclusão das atividades de Iniciação Científica ou Iniciação em desenvolvimento tecnológico e Inovação - Edital 2017/2018 (ano de início e término do Edital).

Prof Dr. Eduardo Todt / Departamento de Informática

Localização de Robôs Móveis em Ambientes Não-estruturados/ 2011025429

CURITIBA
2018

1 Alterações Realizadas no Plano de Trabalho

A plataforma inicial era sobre robótica móvel em Raspberry Pi, mas como o laboratório de Visão, Robótica e Imagem do Departamento de Informática da UFPR conseguiu uma placa de desenvolvimento da NVIDIA, decidiu-se mudar a plataforma para NVIDIA Jetson TX2, com Deep Learning.

2 Resumo

O presente relatório apresenta o andamento do projeto de iniciação científica realizado no laboratório VRI (Visão, Robótica e Imagem) da UFPR sob orientação do prof. Dr. Eduardo Todt. O objetivo da pesquisa é o estudo de Aprendizado de Máquina e comparação de algumas abordagens de Redes Neurais Convolucionais no kit de desenvolvimento integrado NVIDIA Jetson TX2.
Palavras-Chave: Deep Learning, Redes Neurais Convolucionais.

3 Introdução

Os computadores modernos são extremamente rápidos em resolver problemas matemáticos, porém em tarefas mais abstratas como reconhecer objetos em uma imagem, que é tão natural para um humano, pode ser muito complicado para se programar. Inspirando-se no cérebro humano, cientistas reproduziram uma versão simplificada dele em algoritmo. A partir daí, os programas não só solucionam problemas, mas também aprendem a solucioná-los.

Deep Learning é um dos temas mais estudados em Inteligencia Artificial atualmente graças ao seu grande sucesso em tarefas como classificação e segmentação de imagens e em vários outros problemas de Visão Computacional. Esse tema existe há bastante tempo, já em 1943 foi publicado o primeiro modelo computacional de redes neurais artificiais [10], mas só atualmente que essa área se desenvolveu devido as recentes tecnologias em GPU e memória.

Trabalhos em reconhecimento de fala [3], reconhecimento de ação humana suspeita em vigilância [4], segmentação semântica de imagens [9], classificação de imagens usando a base ImageNet [5] e reconhecimento facial [14] mostram que Deep Learning é uma ferramenta poderosa que pode resolver vários dos problemas mais desafiadores da área de Inteligência Artificial.

4 Revisão da Literatura

4.1 Redes Neurais

4.1.1 Visão Geral

Uma rede neural é um emaranhado de neurônios, que geralmente é estruturado em várias camadas com vários neurônios em cada uma. Um neurônio é capaz de aplicar uma função simples, com várias entradas e uma saída, sendo

que essas entradas são as saídas de todos os neurônios da camada anterior (com exceção dos neurônios da primeira camada por ser a camada de entrada), como mostrado na Figura 1.

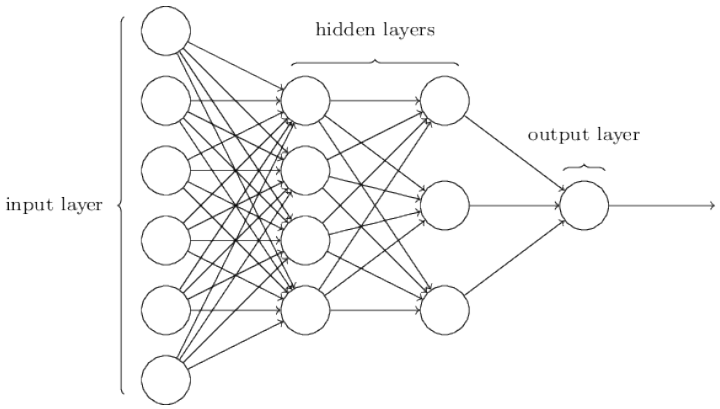

Figura 1: Exemplo de rede neural. [11]

A função aplicada por cada neurônio é a soma de todas as entradas multiplicadas pelos seus respectivos pesos, e depois disso é aplicado uma função no resultado, geralmente sendo a função sigmóide. A função é descrita na seguinte equação: $y_k = \sigma(\sum_{j=0}^{n} w_{j,k} \cdot x_j)$, sendo a saída y de um neurônio k qualquer (y_k) o resultado de uma função sigmóide aplicada em um somatório do produto dos pesos ($w_{j,k}$) pelas saídas (x_j) dos n neurônios da camada anterior. [11]

Dessa forma, os neurônios das primeiras camadas recebem as entradas e propagam os resultados de seus cálculos para a segunda camada. Os neurônios da segunda camada recebem as saídas da primeira, fazem seus cálculos e propagam para a próxima camada, e assim por diante até a camada final que gera as saídas.

4.1.2 Backpropagation

Uma das primeiras aplicações de backpropagation em rede neural foi descrita por David Rumelhart [13], que é o cálculo principal usado no treinamento de redes neurais. O ponto principal do algoritmo de redes neurais está nos valores dos pesos, pois é neles que o aprendizado é realizado. Os pesos são inicializados com valores randômicos e diferentes de zero. Após a propagação em toda a rede, é calculado a taxa de erro em cada saída (como na equação a seguir) e a partir desse valor, a rede é retro-propagada da última camada até a primeira atualizando todos os pesos.

$Erro = (valor\ pretendido - valor\ obtido)^2$.

A cada sessão de treinamento, os pesos são atualizados subtraindo a sua contribuição no erro na classificação. Para isso, calcula-se a derivada parcial do erro em relação ao peso a ser atualizado para se obter a direção em que a função de erro diminui. Essa derivada parcial é multiplicada por um fator de

aprendizagem e então subtraída do peso original, obtendo o novo peso, conforme a equação aplicada em todos os pesos de todos os neurônios: $novo\ w_{j,k} = w_{j,k} - \alpha \cdot \frac{\partial E}{\partial w_{j,k}}$, onde $w_{j,k}$ é o peso w da conexão entre o k-ésimo neurônio e a j-ésimo camada e α é o fator de aprendizagem. [12]

4.2 Redes Neurais Convolucionais

Ao usar uma rede neural totalmente conectada com imagens nas entradas, cada neurônio da camada de entrada corresponderia a cada um dos pixels de uma imagem (n×n). Supondo que uma rede tenha 1 camada de entrada, 2 camadas ocultas de 16 neurônios cada e 1 camada de saída com 10 que trabalhe sobre uma imagem (28×28) sem cores, teríamos um total de $784 \times 16 + 16 \times 16 + 16 \times 10 = 12960$ pesos. Isso mostra que redes neurais totalmente conectadas ficam complexas extremamente rápido a medida que aumentamos o tamanho da imagem.

Contornando esse problema, em 1998 foi publicado um artigo que propôs uma rede neural que popularizou a Rede Neural Convolucional (CNN) [7]. A CNN é a mais apropriada para classificação de imagens por reduzir o número de pesos/parâmetros ao mesmo tempo que aumenta a generalização do método com ideias como pesos compartilhados, campos receptivos locais e subamostragem [6].

As Redes Neurais Convolucionais possuem 4 tipos principais de camadas: Convolutional, Activation, Pooling e Fully-Connected, além da camada de entrada, como mostrado na Figura 2. O grupo de camadas internas (convolução, ativação e pooling) geralmente é repetido algumas vezes ao longo das arquiteturas.

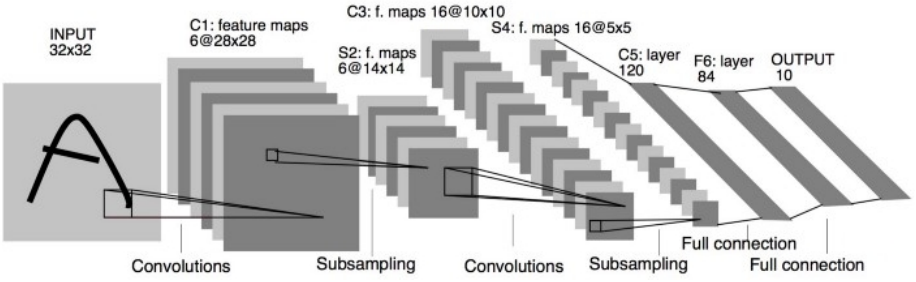

Figura 2: Arquitetura de uma CNN. [6]

- A camada de entrada recebe uma imagem $(m \times n)$. Quando a imagem é em cores RGB, a imagem é de dimensão $(m \times n \times 3)$.
- Na camada de convolução, cada saída é obtida a partir de um produto matricial entre apenas as entradas dentro de uma certa região local da imagem e os seus pesos. Aqui a dimensão de profundidade passa a se

referir ao número de filtros utilizados. Exemplo de saída para uma imagem $(32 \times 32 \times 3)$ após a convolução: $(32 \times 32 \times 12)$, sendo 12 o número de filtros.

- A camada de ativação (também chamada de RELU - Rectified Linear Unit) aplica uma função de ativação em cada elemento da entrada, não alterando as dimensões da camada. Uma função comumente usada é a $max(0, x)$.
- A camada de subamostragem (também chamada de Pooling) aplica uma redução nas dimensões, porém não alterando o número de filtros. Exemplo de saída para uma entrada de $(32 \times 32 \times 12)$: $(16 \times 16 \times 12)$
- A última camada é o classificador da imagem, que computa as taxas de confidência da imagem de entrada para cada categoria e é uma rede neural totalmente conectada.

4.3 Reconhecimento de Plantas

Uma das aplicações de Deep Learning que será discutida nesse artigo é a de reconhecimento de plantas. Em visão computacional, a identificação de plantas é considerado ainda um problema não resolvido já que as plantas na natureza podem ter formatos e cores muito parecidos [8]. Muitas abordagens envolvem a extração de *features* das flores ou folhas a partir do formato ou das nervuras [15].

Essas técnicas citadas anteriormente são muito dependentes do dataset utilizado

5 Materiais e Métodos

Por fins didáticos, utilizamos o DIGITS (Deep Learning GPU Training System) para classificação e teste em deep learning por ser uma interface que reúne vários frameworks e facilita a leitura e separação do banco de imagens em imagens de treinamento e imagens de teste, além de mostrar gráficos com a acurácia ao longo do tempo e também uma fácil visualização das ativações e pesos em cada camada, como mostrado na Figura 3. O DIGITS possui também redes pré-treinadas como LeNet, GoogLeNet e AlexNet.

O laboratório de pesquisa VRI possui duas placas NVIDIA Titan XP de uso coletivo na servidora e um dispositivo embarcado NVIDIA Jetson TX2. Como a Jetson TX2 é um sistema embarcado, não é possível treinar uma rede neural nela pela limitação de memória, por isso a rede deve ser treinada na servidora e posteriormente carregada na Jetson para execução.

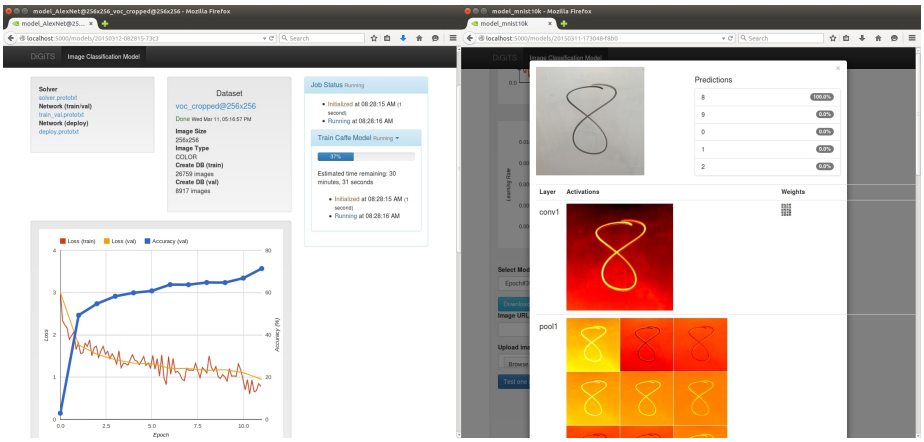

Figura 3: Screenshot de um treinamento de uma CNN usando a interface DIGITS. [1]

6 Resultados e Discussão

Foram feitos experimentos baseados em [2], que classifica imagens do dataset Oxford 102 Category Flower Dataset usando Caffe. Para a inicialização com modelos pré-treinados foi usada a rede AlexNet treinada com a base de dados ILSVRC 2012 ImageNet.

O número de saídas na camada final foi modificada para 102 para melhor refletir o número de categorias do dataset. A taxa de aprendizado global foi diminuída enquanto a taxa para a última camada foi aumentada em relação as camadas anteriores. Após 20.000 iterações, a acurácia foi de 68% no teste com 1020 imagens. O treinamento foi feito tanto na NVIDIA Jetson TX2 quanto na NVIDIA Titan XP e nas imagens 4 e 5 estão os resultados de acurácia e função de erro (loss).

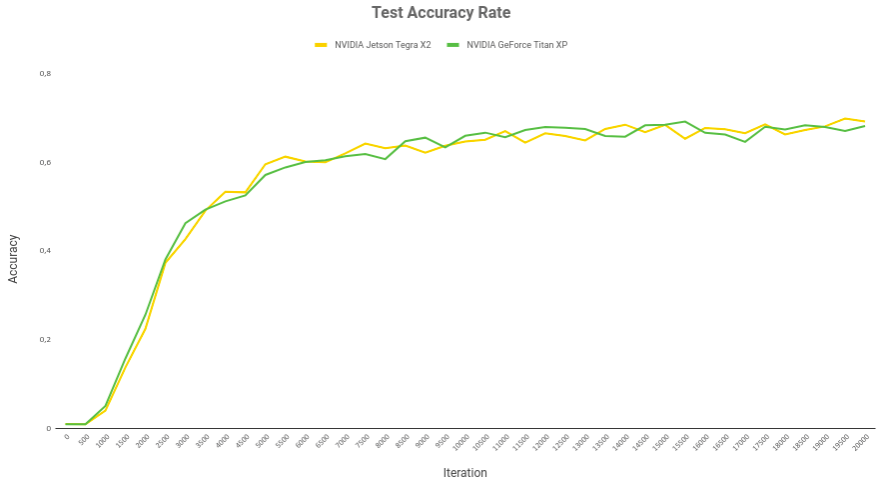

Figura 4: Gráfico da acurácia a cada iteração.

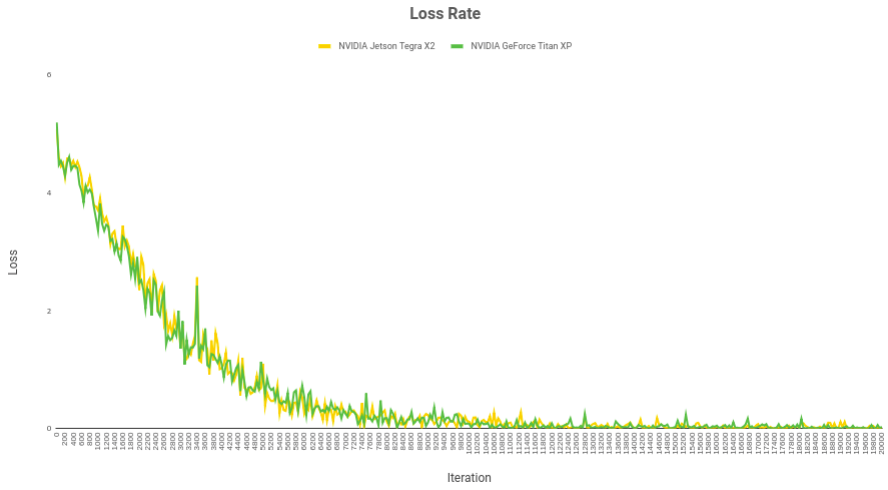

Figura 5: Gráfico da função de perda a cada iteração.

7 Conclusões

Esse trabalho estudou uma abordagem de classificação usando Deep Learning, mais especificamente o reconhecimento de plantas pelas suas flores. Os resultados iniciais do sistema de classificação foram satisfatórios mas não são competitivos em relação a outros métodos usando Deep Learning.

Nosso trabalho futuro focará em examinar com mais detalhes as propriedades da CNN, aumentar o número de iterações e retreiná-la para obter melhores resultados na tarefa de classificação de plantas.

Referências

[1] Dustin Franklin. Guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and Jetson TX1/TX2. `https://github.com/dusty-nv/jetson-inference`. Accessed: 2018-03-22.

[2] Jimmie Goode. caffe-oxford102. `https://github.com/jimgoo/caffe-oxford102`. Accessed: 2018-08-13.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[4] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. Deepplant: Plant identification with convolutional neural networks. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 452–456. IEEE, 2015.

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[10] Warren Mcculloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147, 1943.

[11] Michael A Nielsen. *Neural networks and deep learning*. Determination Press, 2015.

[12] Tariq Rashid. *Make your own neural network*. CreateSpace Independent Publishing Platform, 2016.

[13] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[15] Qingfeng Wu, Changle Zhou, and Chaonan Wang. Feature extraction and automatic recognition of plant leaf using artificial neural network. *Advances in Artificial Intelligence*, 3:5–12, 2006.