

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA
PSI5120 - TÓPICOS EM COMPUTAÇÃO EM NUVEM
PROF.: SERGIO TAKEO KOFUJI

**Explorando o Poder da Computação em Nuvem para o
Aprendizado de máquina usando uma rede neural MLP (Multi
Layer Perceptron) para classificação de objetos em imagens Reais**

SÃO PAULO
2023

Explorando o Poder da Computação em Nuvem para o Aprendizado de máquina usando uma rede neural MLP (Multi Layer Perceptron) para classificação de objetos em imagens Reais

Letícia Pinho da Silva¹ - 7541855
Ruan dos Santos Carvalho ² -12086513

A revolução digital tem provocado uma profunda transformação na maneira como as empresas e organizações conduzem suas operações e tomam decisões estratégicas. Duas tecnologias que têm se destacado nesse cenário são a computação em nuvem e o aprendizado de máquina (Hwang, 2017).

A convergência dessas duas áreas resultou em uma abordagem poderosa que oferece uma série de benefícios e oportunidades para empresas de todos os setores. Neste artigo, exploraremos como a utilização da computação em nuvem e como essa tecnologia tem sido utilizada no campo do aprendizado de máquina.

Computação em Nuvem

Segundo Veras, em seu livro *Cloud Computing - Nova Arquitetura da TI*, define Computação em Nuvem, ou *Cloud Computing*, como um conjunto de recursos virtuais prontamente utilizáveis e acessíveis, incluindo hardware, software, plataformas de desenvolvimento e serviços. Esses recursos têm a capacidade de serem reconfigurados de maneira dinâmica para se adequarem a diferentes níveis de demanda, permitindo uma otimização eficiente de sua utilização. Esse conjunto de recursos é tipicamente acessado através de um modelo de pagamento pelo uso, com garantias fornecidas pelo provedor por meio de acordos de nível de serviço (Veras, 2012).

A essência da Computação em Nuvem envolve a substituição de ativos de tecnologia da informação que normalmente seriam gerenciados internamente por funcionalidades e serviços que são escaláveis de acordo com o crescimento da demanda, tudo isso a preços competitivos de mercado (MARINESCU, 2018). Essas funcionalidades e serviços são desenvolvidos utilizando tecnologias inovadoras, como virtualização, arquiteturas de aplicativos e infra estruturas orientadas a serviços, bem como tecnologias e protocolos baseados na Internet. Isso visa reduzir os custos associados ao hardware e software utilizados para processamento, armazenamento e comunicação de dados (MARINESCU, 2018).

Segundo Veras, as características essenciais da Computação em Nuvem são:

- Autoatendimento sob demanda

¹ Aluna do programa de mestrado em Ciência da Computação do Instituto de Matemática e Estatística da Universidade de São Paulo.

² Aluno do programa de mestrado em Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo.

- Amplo acesso a serviços de rede
- Pool de recursos
- Elasticidade rápida
- Serviços mensuráveis

Aprendizado de máquina

Stuart Russell e Peter Norvig, em seu livro "Artificial Intelligence: A Modern Approach" (Inteligência Artificial: Uma Abordagem Moderna), conceituam a área de aprendizado de máquina como o campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados.

Essa definição destaca a ideia fundamental do aprendizado de máquina, que é permitir que os computadores aprendam a partir de dados e experiências, em vez de serem programados com regras específicas para realizar tarefas. Isso envolve o desenvolvimento de algoritmos e modelos que podem reconhecer padrões nos dados e, com base nesses padrões, fazer previsões ou tomar decisões.

O aprendizado de máquina é amplamente utilizado em uma variedade de aplicações, desde reconhecimento de fala e visão computacional até recomendação de produtos e diagnóstico médico e tem se destacado nos últimos anos. Esse crescimento é impulsionado pelo aumento significativo no volume de dados gerados, bem como pelo avanço da capacidade computacional para a modelagem e processamento dessas informações. Esses avanços têm possibilitado a difusão de técnicas e a aplicação de algoritmos de aprendizado de máquina tanto no âmbito acadêmico quanto no mercado. (RUSSEL e NORVIG, 2013)

Benefícios da utilização da Computação em Nuvem para Aprendizado de Máquina

A utilização de recursos computacionais em nuvem, como servidores, armazenamento, redes e serviços, por meio da internet, permite que as empresas possam acessar e gerenciar recursos sob demanda, aproveitando seus recursos escaláveis e flexíveis para treinar modelos complexos e processar grandes conjuntos de dados (Hwang, 2017).

1. Escalabilidade

Uma das vantagens mais significativas da computação em nuvem é a capacidade de escalar recursos de acordo com a demanda. No contexto do aprendizado de máquina, isso é crucial, pois treinar modelos complexos muitas vezes exige recursos computacionais significativos. Com a nuvem, é possível alocar rapidamente mais capacidade de processamento e armazenamento conforme necessário, acelerando o treinamento de modelos e permitindo lidar com tarefas de maior envergadura.

2. Acesso a Recursos Avançados

As provedoras de serviços em nuvem oferecem uma ampla gama de recursos e serviços avançados, como unidades de processamento gráfico (GPUs) e unidades de

processamento tensorial (TPUs), que são otimizadas para tarefas de aprendizado de máquina. Esses recursos aceleram o tempo de treinamento de modelos e permitem a experimentação com algoritmos mais complexos.

3. Redução de Custos e Tempo

A utilização da nuvem elimina a necessidade de adquirir e manter infraestrutura física, o que pode ser caro e demorado. Além disso, a capacidade de escalar recursos conforme a demanda ajuda a otimizar os custos, uma vez que os recursos podem ser dimensionados para cima ou para baixo de acordo com a necessidade.

4. Colaboração Facilitada

A nuvem também facilita a colaboração entre equipes geograficamente distribuídas. Vários membros da equipe podem acessar e trabalhar nos mesmos recursos de nuvem de qualquer lugar do mundo, promovendo a troca de conhecimento e aprimorando os modelos de machine learning de forma colaborativa.

5. Implantação Simplificada

Após o treinamento, os modelos de machine learning precisam ser implantados para uso em produção. A computação em nuvem oferece soluções de implantação simplificadas, permitindo que os modelos sejam facilmente disponibilizados como serviços web, APIs ou integrações em aplicativos.

Desafios e Considerações

Embora a computação em nuvem ofereça muitos benefícios para o aprendizado de máquina, também apresenta desafios e considerações importantes:

1. Privacidade e Segurança

O uso de serviços em nuvem implica na transferência de dados sensíveis para terceiros. Isso levanta preocupações sobre privacidade e segurança. As organizações precisam garantir que medidas apropriadas de segurança de dados sejam implementadas para proteger as informações confidenciais.

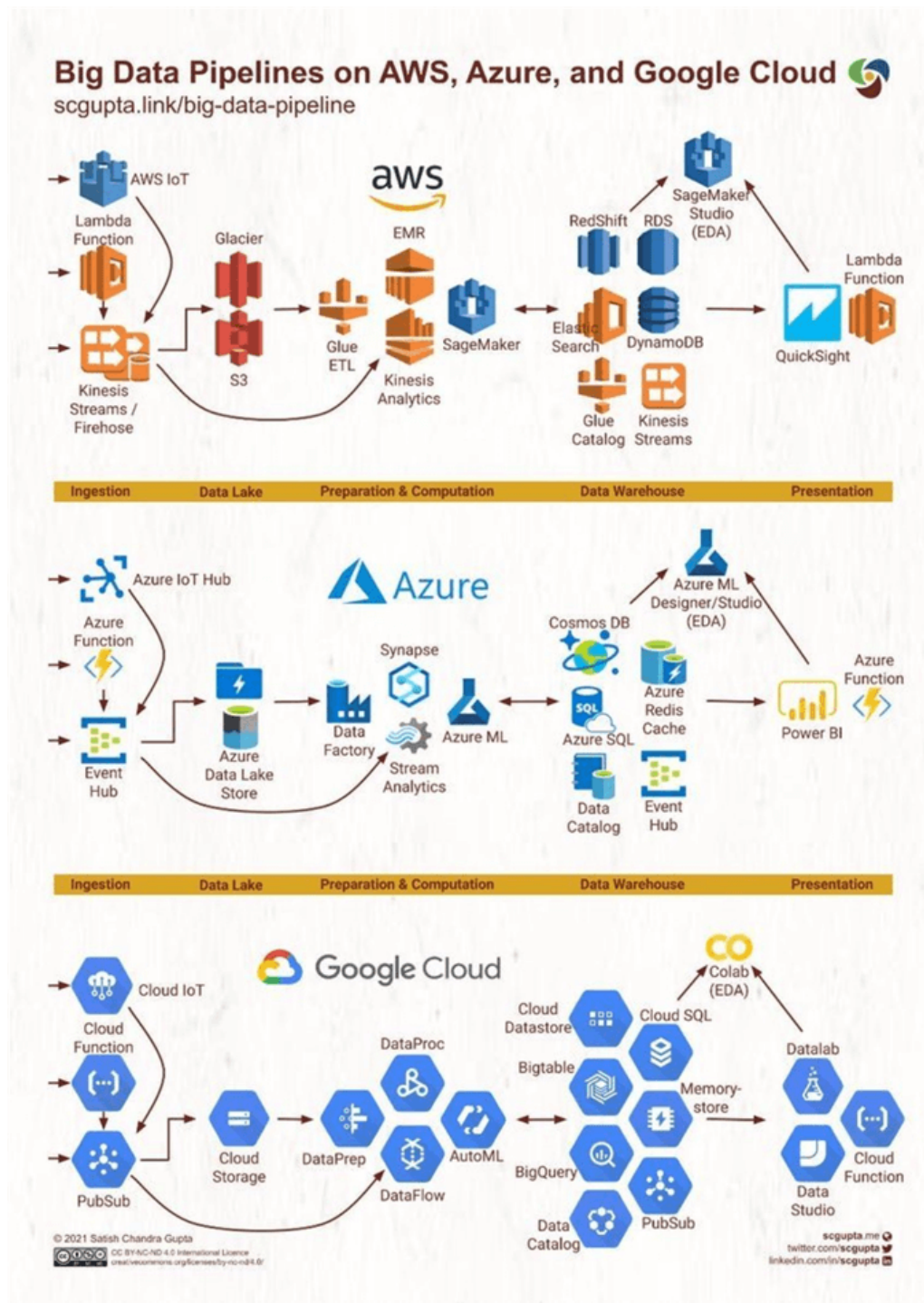
2. Latência e Conectividade

Algoritmos de aprendizado de máquina em nuvem podem depender de uma conexão de internet estável e de baixa latência para funcionar eficazmente. Em cenários onde a conectividade é um problema, a computação em nuvem pode não ser a solução ideal.

3. Custo e Orçamento

Embora a computação em nuvem possa reduzir custos operacionais em muitos casos, é essencial gerenciar cuidadosamente os gastos para evitar surpresas no orçamento. A escalabilidade flexível pode levar a gastos excessivos se não for monitorada adequadamente.

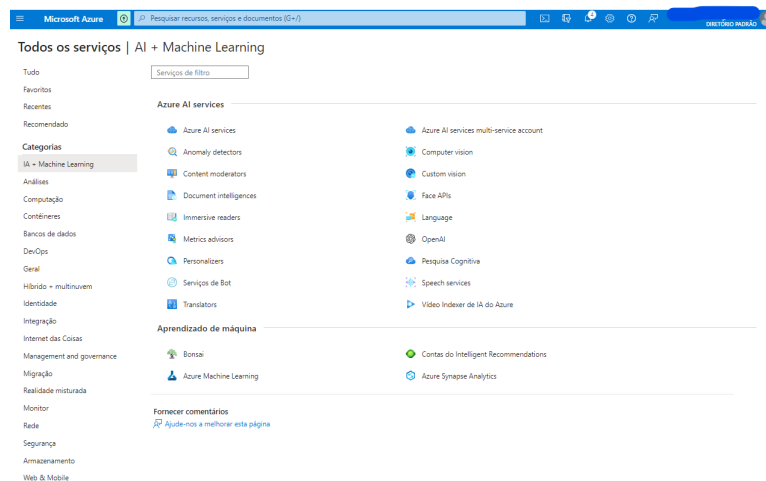
Exemplos de ambientes integrados em Cloud: AWS, Azure e GCP



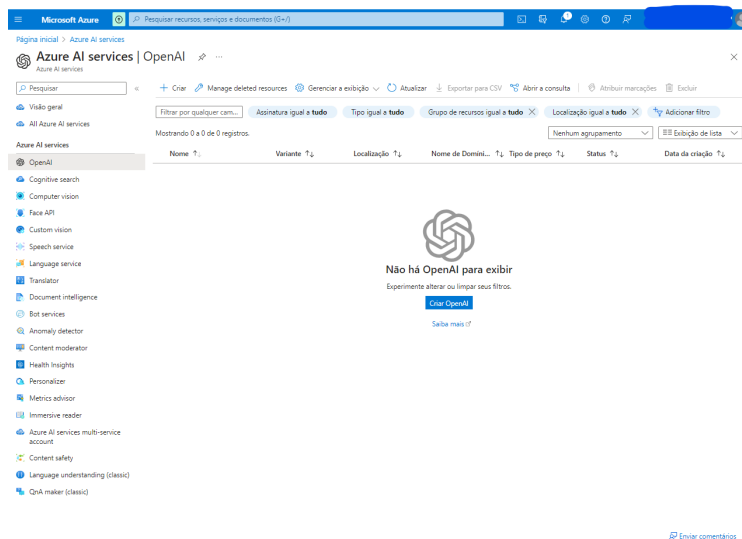
³ Multi-Cloud Data Platform Architecture. Imagem disponível em: <https://www.mssqltips.com/sqlservertip/7316/cloud-data-lakehouse-success-story-architecture-outcomes-lessons-learned/>. Acesso em: 29/08/2023.

Estudo de caso: aprendizado de máquina no Azure

O Azure possui uma plataforma colaborativa de análise de big data baseada na nuvem, projetada para acelerar a implantação de projetos de big data e machine learning⁴. Além dos serviços mais gerais, onde é possível configurar máquinas para implementar os algoritmos, também é possível utilizar o Azure Databricks. Essa plataforma foi desenvolvida em parceria pela Microsoft e pela Databricks, e oferece um ambiente unificado para cientistas de dados, engenheiros e analistas colaborarem na construção, treinamento e implantação de modelos de aprendizado de máquina. Integrando-se perfeitamente com serviços Azure, como o Azure Machine Learning e o Azure Data Lake, o Azure Databricks.



5



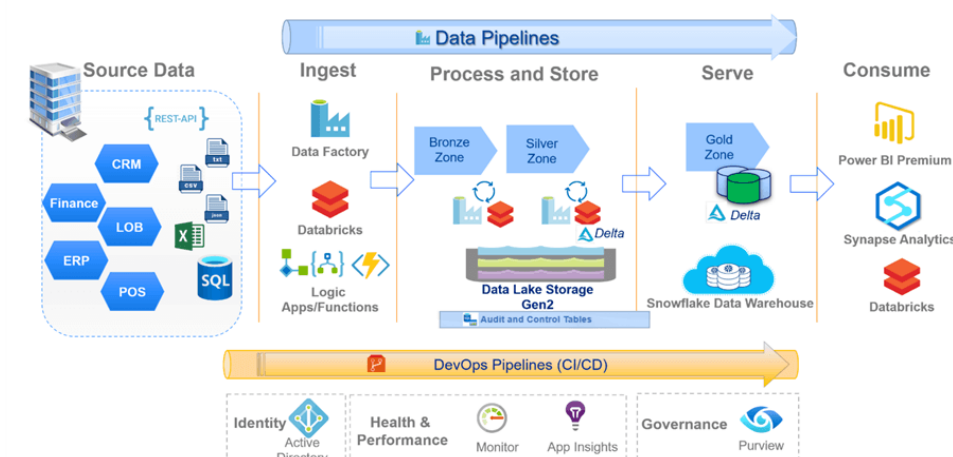
6

⁴ Azure Machine Learning. Disponível em:<<https://azure.microsoft.com/pt-br/products/machine-learning>>. Acesso em: 27/08/2023.

⁵ Serviços de IA e Machine Learning. Imagem retirada do painel de serviços da Azure.

⁶ Ferramentas dos serviços integrados com a OpenAI. Imagem retirada do painel de serviços da Azure.

The Azure Data Lakehouse Platform



7

No que diz respeito à característica da Escalabilidade Sob Demanda, o Azure Databricks permite que as equipes dimensionem seus recursos de acordo com as necessidades do projeto. Isso é fundamental para o aprendizado de máquina, onde o treinamento de modelos complexos pode exigir considerável capacidade computacional. A escalabilidade sob demanda do Azure Databricks agiliza o processo de treinamento, reduzindo o tempo necessário para concluir projetos.

Assim como outras empresas do mercado, como a AWS e a GCP, a plataforma se integra perfeitamente com outros serviços do ecossistema Azure. O Azure Databricks também simplifica o gerenciamento de clusters e recursos de computação. Isso libera as equipes de tarefas de manutenção, como ajuste de recursos e gerenciamento de servidores. Vale ressaltar que, assim como com qualquer plataforma em nuvem, é crucial gerenciar os custos de maneira eficaz. O uso indiscriminado de recursos pode resultar em gastos excessivos, tornando necessário monitorar e otimizar regularmente a utilização da plataforma.

Em relação à exploração de dados e visualização, a plataforma oferece uma variedade de ferramentas que facilitam a análise e a compreensão dos conjuntos de dados. Isso é essencial para a etapa de pré-processamento de dados e para a identificação de padrões relevantes para o treinamento dos modelos. A interface é intuitiva mas ainda pode resultar em uma curva de aprendizado íngreme para equipes que não estão familiarizadas com as tecnologias envolvidas.

⁷ Azure Data Platform Architecture. Imagem disponível em: <https://www.mssqltips.com/sqlservertip/7316/cloud-data-lakehouse-success-story-architecture-outcomes-lessons-learned/>. Acesso em: 29/08/2023.

Experimento prático: utilização da azure cloud para rede treinar rede neural densa para classificação

É uma aplicação de técnicas de aprendizado de máquina para resolver um problema de classificação de imagens. O conjunto de dados Fashion MNIST consiste em 60.000 imagens de 10 categorias diferentes de roupas, com 6.000 imagens por categoria. Cada imagem é uma representação em escala de cinza de 28x28 pixels.

O uso de uma rede neural densa (também conhecida como feedforward ou fully connected neural network) para resolver esse problema se justifica por algumas razões teóricas:

1ª - Simplicidade e Intuição Inicial: Redes neurais densas são o tipo mais básico de redes neurais e servem como uma introdução lógica para entender os conceitos fundamentais de redes neurais. Elas consistem em camadas de neurônios totalmente conectados, onde cada neurônio em uma camada está conectado a todos os neurônios da camada anterior.

2ª - Aprendizado de Recursos Hierárquicos: Embora as redes neurais densas não explorem a estrutura espacial das imagens como as redes convolucionais, elas ainda são capazes de aprender representações hierárquicas de características. À medida que as informações passam pelas camadas, a rede pode aprender a combinação de características mais simples em características mais complexas, auxiliando na classificação.

3ª - Facilidade de Implementação em Keras: A biblioteca Keras, integrada ao TensorFlow, permite a construção de redes neurais de forma intuitiva e rápida. Com algumas linhas de código, é possível construir, treinar e avaliar uma rede neural densa para classificação.

No entanto, é importante notar que redes neurais densas podem não ser a abordagem mais sofisticada ou eficaz para a classificação de imagens, especialmente em conjuntos de dados grandes e complexos como o Fashion MNIST. Redes convolucionais (CNNs) geralmente superam as redes neurais densas em tarefas de visão computacional devido à sua capacidade de capturar padrões locais e espaciais nas imagens.

Aqui está uma representação visual simples de uma arquitetura de rede neural densa para a classificação do Fashion MNIST:

Model: "sequential_9"		
Layer (type)	Output Shape	Param #
=====		
flatten_9 (Flatten)	(None, 784)	0
dense_29 (Dense)	(None, 256)	200960
dropout_2 (Dropout)	(None, 256)	0
dense_30 (Dense)	(None, 128)	32896
dropout_3 (Dropout)	(None, 128)	0
dense_31 (Dense)	(None, 64)	8256
dense_32 (Dense)	(None, 32)	2080
dense_33 (Dense)	(None, 10)	330
=====		
Total params: 244,522		
Trainable params: 244,522		
Non-trainable params: 0		

Esse modelo é uma representação resumida de uma rede neural sequencial denominado "sequential_9". O modelo possui várias camadas empilhadas em sequência para realizar uma tarefa de classificação. Aqui está um resumo da arquitetura do modelo:

Camada de Entrada (Flatten):

Tipo: Flatten

Saída: (None, 784)

Descrição: Essa camada converte uma imagem bidimensional em uma matriz unidimensional de 784 elementos (28x28 pixels).

Camada Densa (Fully Connected):

Tipo: Dense

Saída: (None, 256)

Parâmetros: 200,960

Descrição: Camada densa com 256 neurônios. Cada neurônio está conectado a todos os neurônios da camada anterior.

Camada Dropout:

Tipo: Dropout

Saída: (None, 256)

Descrição: Camada de regularização que desativa aleatoriamente alguns neurônios durante o treinamento para evitar overfitting. Nenhuma informação passa por essa camada durante a inferência.

Camada Densa (Fully Connected):

Tipo: Dense

Saída: (None, 128)

Parâmetros: 32,896

Descrição: Camada densa com 128 neurônios.

Camada Dropout:

Tipo: Dropout

Saída: (None, 128)

Descrição: Camada de regularização.

Camada Densa (Fully Connected):

Tipo: Dense

Saída: (None, 64)

Parâmetros: 8,256

Descrição: Camada densa com 64 neurônios.

Camada Densa (Fully Connected):

Tipo: Dense

Saída: (None, 32)

Parâmetros: 2,080

Descrição: Camada densa com 32 neurônios.

Camada Densa (Saída):

Tipo: Densa

Saída: (None, 10)

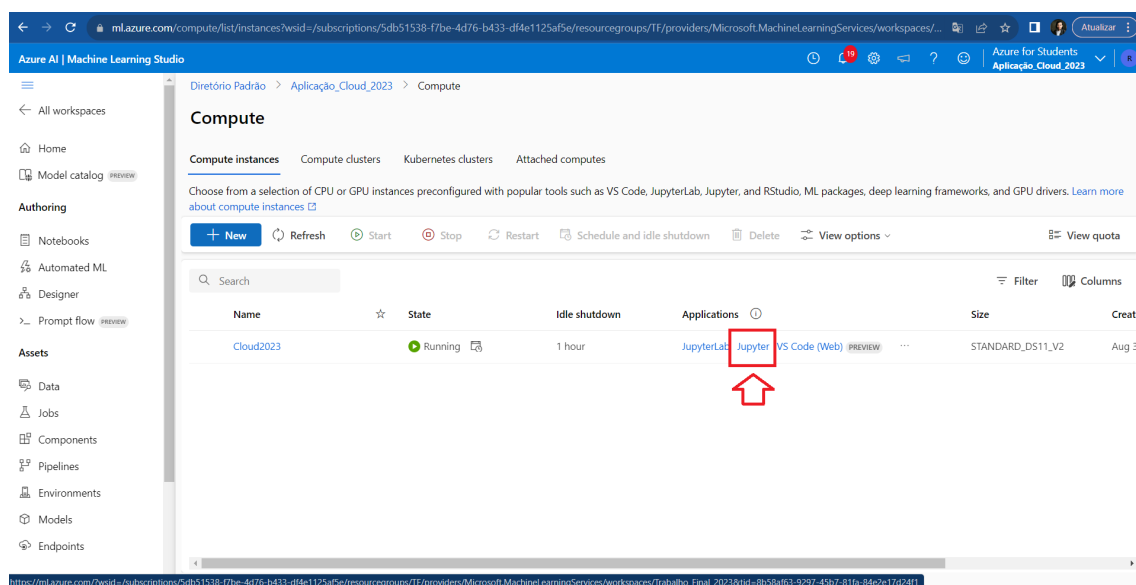
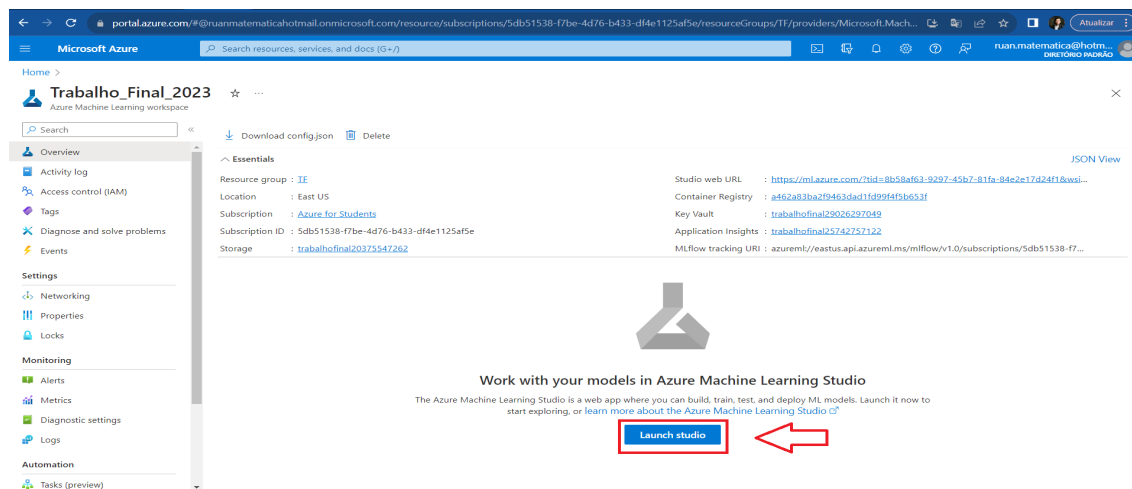
Parâmetros: 330

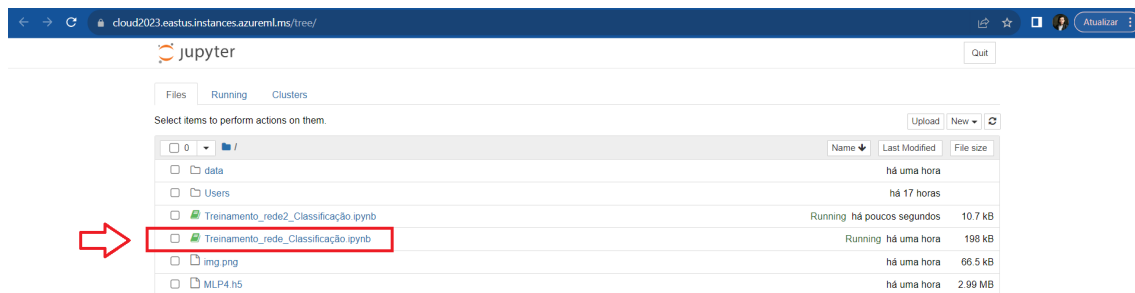
Descrição: Camada de saída com 10 neurônios, correspondendo às 10 classes de roupas no conjunto de dados Fashion MNIST.

Total de Parâmetros: 244,522

O modelo é construído em uma configuração sequencial, onde cada camada se comunica diretamente com a próxima na sequência. O objetivo provável desse modelo é realizar a classificação de imagens de roupas em 10 categorias diferentes usando as informações extraídas pelas camadas densas. O uso de camadas de dropout ajuda a regularizar o modelo e evitar overfitting durante o treinamento.

Etapa de Treinamento





```

In [12]: # Importações
import os
import tensorflow.keras as keras
from keras.datasets import fashion_mnist
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout
from keras.utils import to_categorical
from keras import optimizers
import matplotlib.pyplot as plt
import numpy as np
import sys

# Carregamento dos dados
(XA, AY), (XQ, QY) = fashion_mnist.load_data()
XA = 255 - XA
XQ = 255 - XQ

n_classes = 10
AY2 = to_categorical(AY, n_classes)
QY2 = to_categorical(QY, n_classes)

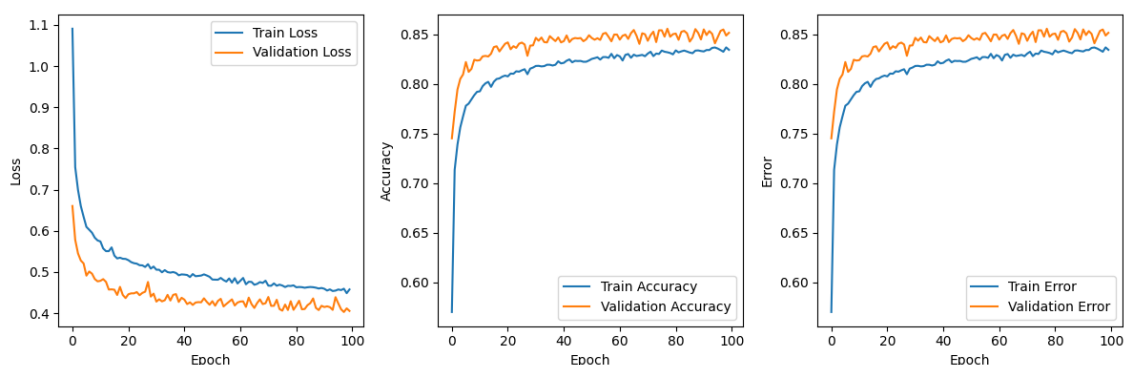
n1, n2 = XA.shape[1], XA.shape[2] # 28, 28
XA = XA.astype('float32') / 255.0 # 0 a 1
XQ = XQ.astype('float32') / 255.0 # 0 a 1

# Definição do modelo
model = Sequential()
model.add(Flatten(input_shape=(n1, n2)))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(n_classes, activation='softmax'))

# Resumo do modelo

```

Métricas de avaliação

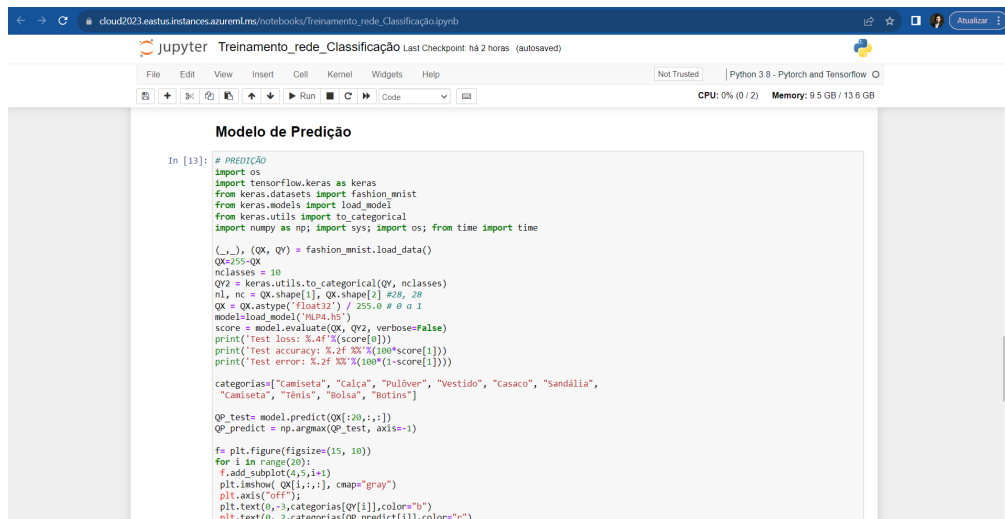


Test loss: 0.4053678512573242

Test accuracy: 85.17 %

Test error: 14.83

Modelo de Predição



```
In [13]: # PREDIÇÃO
import os
import tensorflow.keras as keras
from keras.datasets import fashion_mnist
from keras.models import load_model
from keras.utils import to_categorical
import numpy as np; import sys; import os; from time import time

(X_train, QX, QY) = fashion_mnist.load_data()
QX = QX[:255]
n_classes = 10
QY2 = keras.utils.to_categorical(QY, n_classes)
n1, n2 = QX.shape[1], QX.shape[2] # 28, 28
QX = QX.astype('float32') / 255.0 # 0 a 1
model = load_model('MLP4.h5')
score = model.evaluate(QX, QY2, verbose=False)
print('Test loss: %.4f' % (score[0]))
print('Test accuracy: %.2f %%%' % (100 * score[1]))
print('Test error: %.2f %%%' % (100 * (1 - score[1])))

categorias = ["Camiseta", "Calça", "Pulôver", "Vestido", "Casaco", "Sandália",
              "Camiseta", "Tênis", "Bolsa", "Botins"]

QY_test = model.predict(QX[:20,:])
QY_predict = np.argmax(QY_test, axis=-1)

fig = plt.figure(figsize=(15, 10))
for i in range(20):
    fig.add_subplot(4,5,i+1)
    plt.imshow(QX[i,:], cmap="gray")
    plt.axis('off');
    plt.text(0,-3,categorias[QY[i]],color="b")
    plt.text(0,-2,categorias[QY_predict[i]],color="r")
```

O código fornecido realiza a predição de roupas do conjunto de dados Fashion MNIST utilizando um modelo de rede neural treinado previamente. Aqui está um resumo do que o código faz passo a passo:

1. Importa as bibliotecas necessárias:

- 'os' para funcionalidades relacionadas ao sistema operacional.
- 'tensorflow.keras' para construção e manipulação de modelos de rede neural.
- 'numpy' para operações numéricas eficientes.
- 'sys' para interações com o sistema.
- 'time' para medição de tempo.

2. Carrega o conjunto de dados Fashion MNIST, dividindo-o em conjuntos de treinamento e teste. Também faz uma transformação nos dados de entrada para inverter a escala de cores.

3. Define o número de classes no conjunto de dados (que é 10, representando diferentes tipos de roupas).

4. Converte os rótulos do conjunto de teste em codificação one-hot usando a função 'to_categorical' do Keras.

5. Obtém as dimensões dos exemplos de entrada do conjunto de teste.

6. Normaliza os dados de entrada do conjunto de teste para ter valores entre 0 e 1.

7. Carrega um modelo pré-treinado chamado 'MLP4.h5'.

8. Avalia o modelo carregado nos dados de teste, calculando a perda e a acurácia. Os resultados são impressos na saída.

9. Define uma lista de categorias de roupas correspondentes aos rótulos das classes.

10. Faz previsões para os primeiros 20 exemplos do conjunto de testes usando o modelo carregado.

11. Para cada exemplo de teste:

- Gera um gráfico para mostrar a imagem do item de vestuário.
- Mostra o nome da categoria verdadeira em azul.
- Mostra o nome da categoria prevista em vermelho.

12. Salva o gráfico gerado como "img.png" e o exibe na saída.

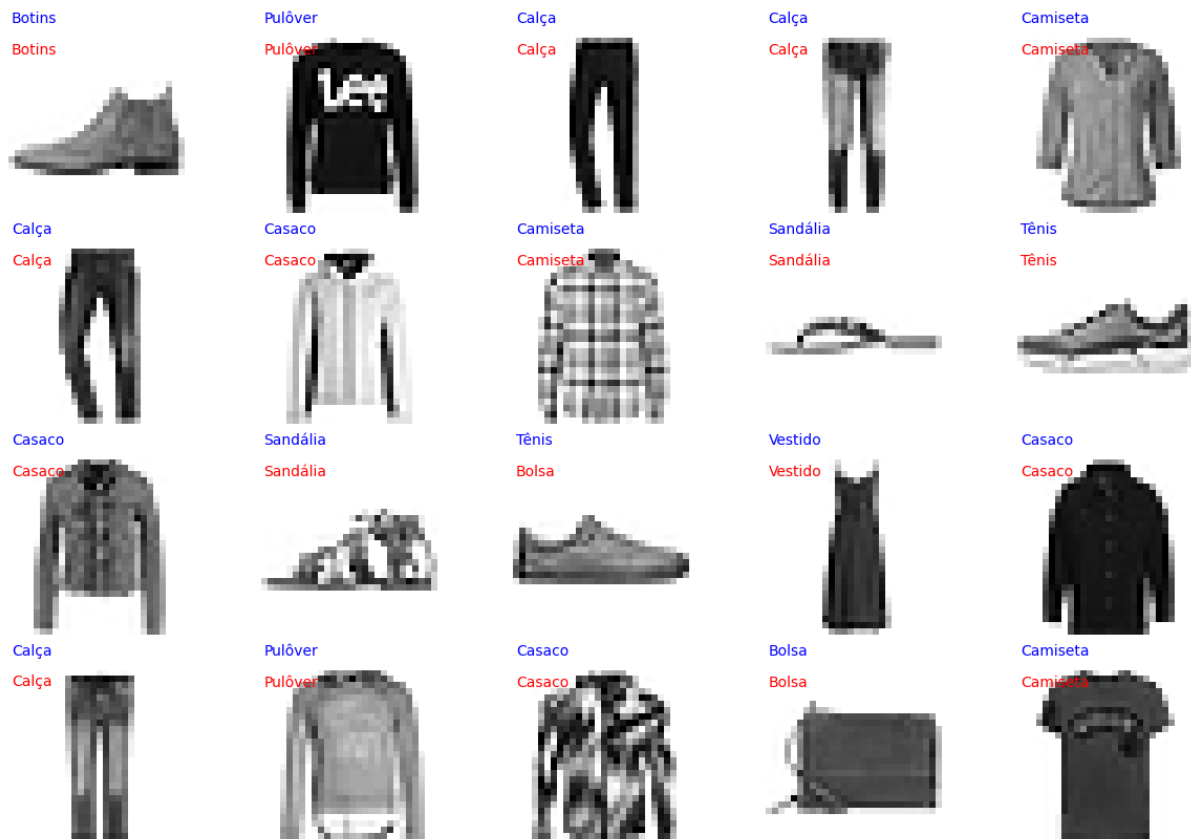
Resultados dos testes

Test loss: 0.4054

Test accuracy: 85.17%

Test error: 14.83%

Esses resultados indicam que o modelo tem uma precisão de cerca de 85.17% na classificação das imagens de roupas do conjunto de teste do Fashion MNIST. A taxa de erro de teste é de aproximadamente 14.83%.



Em resumo, a rede neural densa para a classificação do Fashion MNIST apresentou uma precisão de teste razoável de 85.17%, o que indica que o modelo está fazendo um bom trabalho na classificação das imagens, mas ainda há espaço para melhorias. Isso porque quanto mais próxima a precisão estiver de 100%, melhor o desempenho do modelo.

A perda de teste de 0.4054 também sugere que o modelo está minimizando o erro de forma eficaz. O erro de teste de 14.83% mostra a taxa de classificação incorreta, e reduzir esse valor seria um objetivo importante para melhorar ainda mais o desempenho do modelo e é uma sugestão para trabalhos futuros.

O código utilizado no experimento pode ser acessado na íntegra do repositório do projeto disponível em: https://github.com/ruan-math/Rede_Neural_MLP

Conclusão

A combinação da computação em nuvem e do aprendizado de máquina está transformando a maneira como as organizações abordam a análise de dados, tomam decisões e desenvolvem produtos e serviços inovadores. Ao aproveitar a escalabilidade, os recursos avançados e a flexibilidade oferecidos pela nuvem, as empresas podem acelerar o desenvolvimento de modelos de machine learning e explorar novos horizontes no campo da inteligência artificial.

No entanto, é importante abordar cuidadosamente os desafios e considerações para maximizar os benefícios dessa abordagem e garantir a segurança e a eficácia das soluções implementadas.

REFERÊNCIAS

HWANG, K. **Cloud Computing for Machine Learning and Cognitive Applications**. Cambridge, MA : The MIT Press, 2017.

L'ESTEVE, R. C. Machine Learning in Databricks. In: The Definitive Guide to Azure Data Engineering. Apress, Berkeley, CA. 2021. Disponível em: <https://link.springer.com/chapter/10.1007/978-1-4842-7182-7_23>. Acesso em: 25/08/2023.

MENG et al. MLlib: Machine Learning in Apache Spark. Journal of Machine Learning Research. Disponível em: <<https://www.jmlr.org/papers/volume17/15-237/15-237.pdf>>. Acesso em: 25/08/2023.

RAGHAVENDR, K. R., ELGARAL, A. Low-Code Machine Learning Platforms: A Fastlane to Digitalization. Luleå University of Technology. Disponível em: <<https://www.mdpi.com/2227-9709/10/2/50>>. Acesso em: 23/08/2023.

RUAN, W., CHEN, Y., FOROURAGHI, B. On Development of Data Science and Machine Learning Applications in Databricks. Lecture Notes in Computer Science, 2019. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-23381-5_6>.

VERAS, M. Cloud Computing: Nova Arquitetura da TI. Editora Brasport. Rio de Janeiro, 2012.

ZAHARIA, M. Designing production-friendly machine learning. Stanford and Databricks. Disponível em: <<https://dl.acm.org/doi/abs/10.14778/3484224.3484241>>. Acesso em: 26/08/2023.