

**UNIVERSIDADE FEDERAL DE SÃO PAULO
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA E TECNOLOGIA**

**Projeto 05:
Autoencoders**

Nome: Willian Dihanster Gomes de Oliveira **RA:** 112269

**SÃO JOSÉ DOS CAMPOS
2018**

Projeto 5 - Autoencoder

Implementar um autoencoder(usar Keras)

Datasets: MNIST* & AR Face Database**

Variar o número de neurônios internos e o número de camadas (stacks)

MNIST Dataset

- **Experimentos Autoencoders Simples**

Com o script de autoencoder simples (1 stacks) encontrado no site do Keras e apenas aumentando o número de épocas para 100, temos os seguinte resultados da Figura 1, loss: 0.0972 e val_loss: 0.0957

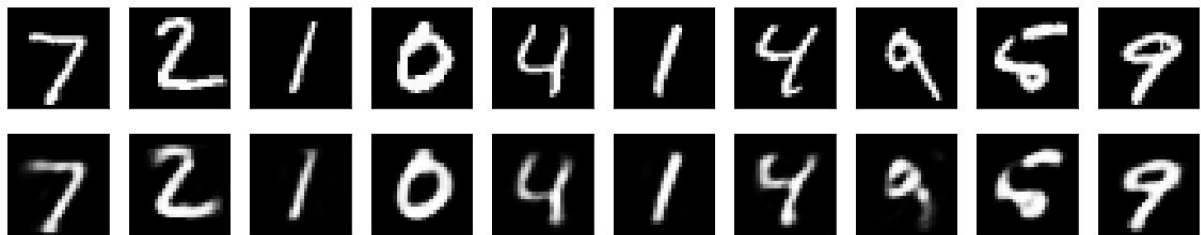


Figura 1: Dígitos reconstruídos pela técnica de autoencoder simples.

Pode-se notar que foi possível representar os números suficientemente bem. Apenas alguns exemplos, como o que representa o número 9, houve um leve “borrado”.

- **Experimentos Deep Autoencoders**

Já com os script para deep autoencoders encontrado no site do Keras e com os parâmetros padrão de configuração das camadas do script (Figura 1), temos como resultado loss: 0.0989, val_loss: 0.0980 e a Figura 3.

```
# "encoded" is the encoded representation of the inputs
encoded = Dense(encoding_dim * 4, activation='relu')(input_img)
encoded = Dense(encoding_dim * 2, activation='relu')(encoded)
encoded = Dense(encoding_dim, activation='relu')(encoded)

# "decoded" is the lossy reconstruction of the input
decoded = Dense(encoding_dim * 2, activation='relu')(encoded)
decoded = Dense(encoding_dim * 4, activation='relu')(decoded)
decoded = Dense(784, activation='sigmoid')(decoded)
```

Figura 3: Parâmetros padrão para as camadas.

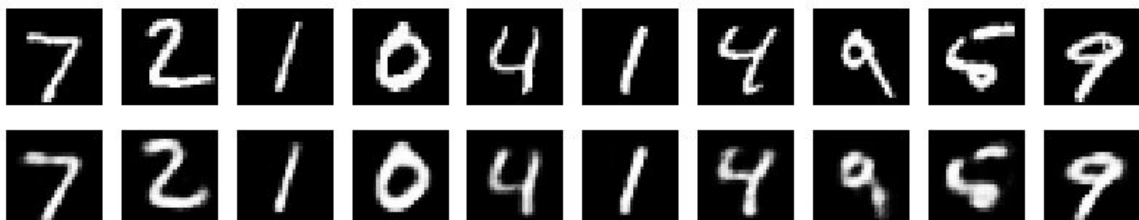


Figura 3: Dígitos reconstruídos pela técnica de deep autoencoder.

Fazendo uma comparação com os resultados para um autoencoder simples, pode-se perceber que os novos resultados foram ligeiramente superiores em termos da qualidade da representação e também menores valores de loss e val_loss.

- **Varição Neurônios Internos**

Então, foram feitos novos experimentos variando o neurônios de neurônios internos e agora, mantendo o número de *stacks*, conforme os parâmetros da Figura 4, temos os resultados da Figura 5 e loss: 0.1021 e val_loss: 0.1035.

```
# "encoded" is the encoded representation of the inputs
encoded = Dense(encoding_dim * 3, activation='relu')(input_img)
encoded = Dense(encoding_dim * 2, activation='relu')(encoded)
encoded = Dense(encoding_dim, activation='relu')(encoded)

# "decoded" is the lossy reconstruction of the input
decoded = Dense(encoding_dim * 2, activation='relu')(encoded)
decoded = Dense(encoding_dim * 3, activation='relu')(decoded)
decoded = Dense(784, activation='sigmoid')(decoded)
```

Figura 4: Parâmetros reconfigurados para as camadas.



Figura 5: Dígitos reconstruídos pela técnica de deep autoencoder com novos parâmetros.

Pode-se notar que para esse aumento de número de neurônios internos, houve mais confusão na representação do número 9, além de um valor ligeiramente, mais alto de loss e val_loss.

Agora, configurando como os parâmetros da Figura 6, (com o dobro do número de neurônios internos em relação ao script original) temos como resultado loss: 0.0911 - val_loss: 0.0906 e a Figura 7.

```
# "encoded" is the encoded representation of the inputs
encoded = Dense(encoding_dim * 8, activation='relu')(input_img)
encoded = Dense(encoding_dim * 4, activation='relu')(encoded)
encoded = Dense(encoding_dim, activation='relu')(encoded)

# "decoded" is the lossy reconstruction of the input
decoded = Dense(encoding_dim * 4, activation='relu')(encoded)
decoded = Dense(encoding_dim * 8, activation='relu')(decoded)
decoded = Dense(784, activation='sigmoid')(decoded)
```

Figura 6: Parâmetros reconfigurados para as camadas.



Figura 7: Dígitos reconstruídos pela técnica de deep autoencoder com novos parâmetros.

Notou-se que para esses experimentos, houve uma melhora na representação dos dígitos, com o número 9 ficando mais fiel ao original e também um menor valor de loss e val_loss. Ou seja, resultados se mostraram superiores em relação à todos os outros experimentos. Sendo assim, mais neurônios podem ajudar na obtenção de uma melhor representação.

- **Dimensão de *Encoding***

Esses experimentos anteriores consideram *encoding_dim* = 32, então novos experimentos foram feito tentando ver as diferenças com outros valores de dimensões para o enconder. Com *encondig_dim* = 16 (metade do original) temos os resultados da Figura 8, loss: 0.1237 val_loss: 0.1236



Figura 8: Dígitos reconstruídos pela técnica de deep autoencoder com *encondig_dim* = 16.

Com *encondig_dim* = 64 (dobro do original) temos os resultados da Figura 9, loss: 0.0886 - val_loss: 0.0883.



Figura 8: Dígitos reconstruídos pela técnica de deep autoencoder com *encondig_dim* = 64.

Pode-se perceber que como esperado, valores menores de *encondig_dim* apresentam mais dificuldades na representação dos dígitos, enquanto o contrário com valores maiores de *encondig_dim*. No entanto, ambos resultados podem ser considerados bons. Pois com cerca de 1/3 da informação original, conseguiu-se representar razoavelmente bem os dados originais.

Conclusões

Com esses experimentos foi possível que a técnica de deep autoencoders apresentou ligeiramente melhores resultados em relação aos autoencoders simples, mas ambas técnicas representaram o conjunto de dados de forma consideravelmente bem, com uma dimensão bem menor que a original. Verificando assim, a eficiência das técnicas de autoencoders e deep autoencoders na diminuição de dimensionalidade.