# Deep Learning Cheat Sheet

## Data Preparation

**Min-max [0,1]** : $x' = \frac{(x - x_{min})}{(x_{max} - x_{min})}$

**Min-max [-1,1]** : $x' = 2 \cdot min\_max(x) - 1$

min-max doesn't handle outliers.

**Z-norm** : $x' = \frac{(x - \mu)}{\sigma}$

**Scaling & Centering**

Scaling improves the numerical stability, the convergence speed and accuracy of the learning algorithms. Centering improves the robustness of the learning algorithms

## Activation Functions

**Sigmoid** : $f(z) = \frac{1}{1+e^{-z}}$ — Smooth and differentiable. Used in output layers for binary classification.

**Hyperbolic Tangent (tanh)** : $f(z) = \tanh(z)$ — Smooth, differentiable, output centered around 0. Used in LSTM.

**Rectified Linear Unit (ReLU)** : $f(z) = \max(0, z)$ — Non-linear, used as a standard, but has dying units problem for $z < 0$.

**Leaky ReLU** : $f(z) = \begin{cases} z & \text{if } z \geq 0 \\ \alpha z & \text{if } z < 0 \end{cases}$ — Addresses dying units problem with a small $\alpha$ (typical $\alpha = 0.01$).

**Exponential Linear Unit (ELU)** : $f(z) = \begin{cases} z & \text{if } z \geq 0 \\ \alpha(e^z - 1) & \text{if } z < 0 \end{cases}$ — Similar to Leaky ReLU but more computationally expensive.

**Softmax** : $f(z_i) = \frac{e^{z_i}}{\sum_{j=0}^{K-1} e^{z_j}}$ — Used in the last layer for multi-class classification, outputs a probability distribution.

## Gradient Descent

**MSE**
**Cross Entropy**
**Stochastic vs. Minibatch**

## Performance Measures

**Matrice de Confusion**
**Confusion Table**
**ROC**
**Precision**
**Recall**

## Bias & Variance

**Model Selection**

## Theory

**Compute Graph**
**Universal Approximation Theorem**

## Curse of Dimensionality

## Backpropagation

**MLP Layer**
**Matrix Notation**
**Full Batch**
**Batch Normalization**

## Vanishing Exploding Gradient

**Saturation**
**Variance Change**
**Xavier & Heu Initialization**
**Batch Normalization**
**Non Saturating Activation Function**
**Gradient Clipping**

## Optimizers

**Momentum**
**AdaGrad**
**RMS Prop**
**Adam**
**Scheduler**

## Regularization

**Weight Penalty**
**Dropout**
**Early Stopping**

## CNN

**Convolutional Layer**
**Pooling Layer**

## Unbalanced Dataset

**Bayesian Approach**
**Discrete**
**Continuous**
**Medical Test**

## DeepCNN

**Conf2D Params**
**MaxPooling**
**LeNet5**
**AlexNet**
**VGGnet**
**GoogleNet**
**ResNet**
**Pattern**

## Feature Visualization

**Data Preparation**
**Network**
**Compile**
**Evaluate**
**Activation Map**

## Data Augmentation

**Principle**
**Types**
**Strategies**
**Keras**

## Functional API

**Sequential vs Functionals**
**Architecture 1**
**Architecture 2**
**Architecture 3**

## Transfer Learning

**Principle**
**Keras Code**
**MobileNet**
**Strategies**

## RNN

**Use Case**
**Model Category**
**Recurrence Net**
**Single Layer**
**Many to Many**
**Un exemple par catégorie**
**Stacked RNN**

## LSTM

**Long Term Memory Unit Cell**
**Gates**
**Backprop**
**Keras**
**GRE**

## Word Embedding

**Word**
**Training**

## Sentiment Classification

**Strategy**
**Architecture**

## Autoencoder

**Definition**
**Use Case**

## GenRNN

**Many to Many**
**Many to One**

## Attention

**Sequence to Sequence**
**Attention**

## Transformer

**High-Level Architecture**
**Self-Attention**
**Full Architecture**