

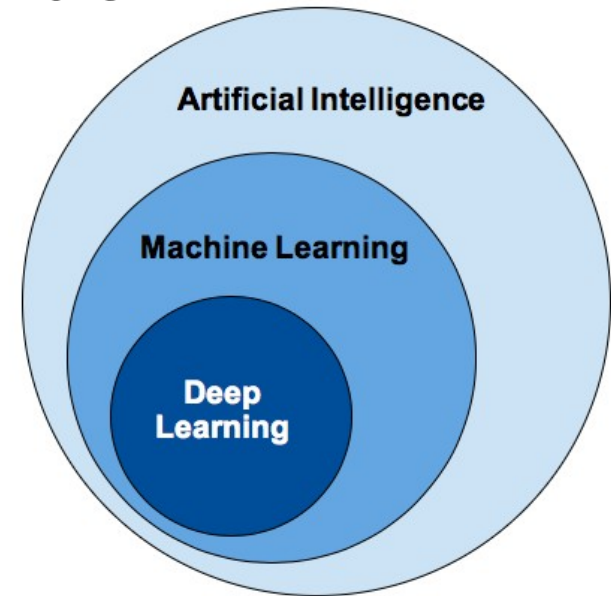
# **Tensorflow et Keras pour le Deep Learning (Apprentissage profond)**

**Benkedadra Mohamed  
Benkorreche Mohammed El Amine  
Youcefi Mohammed Yassine**

**Groupe 1 – ISI – M1**

# Deep Learning

- Fait partie du Machine Learning
- Inspiré par le cerveau humain
- Appliqué sur :
  - Vision Artificielle
  - Reconnaissance Automatique de la parole
  - Traitement Du Langage Naturel
  - ... etc



# Deep Learning

$$h(\text{attr}) = \theta_0 + \theta_1 * \text{attr}_1 + \theta_2 * \text{attr}_2 + \dots + \theta_n * \text{attr}_n$$

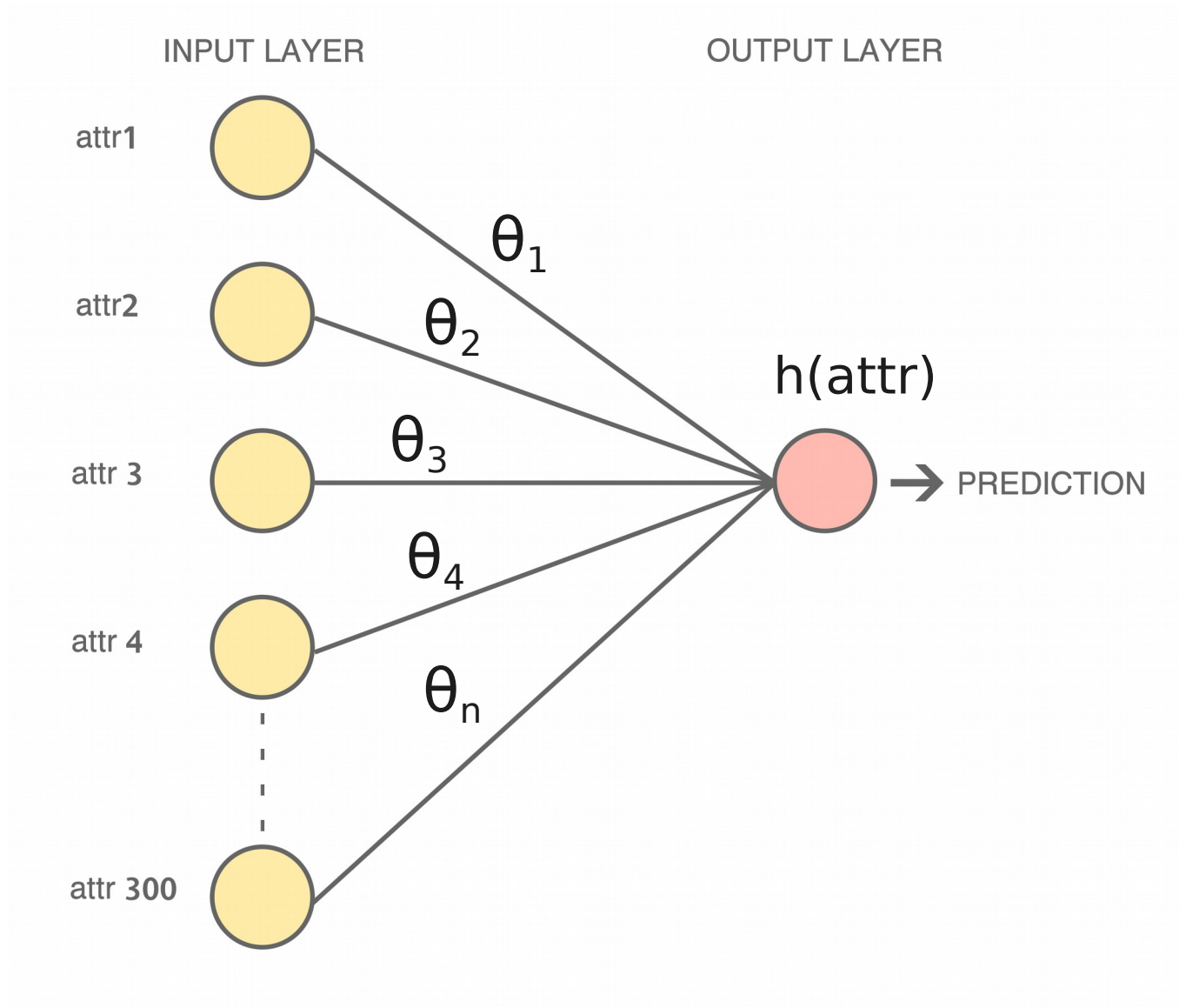
$h$  Hypothèse

$\theta_n$  Poids

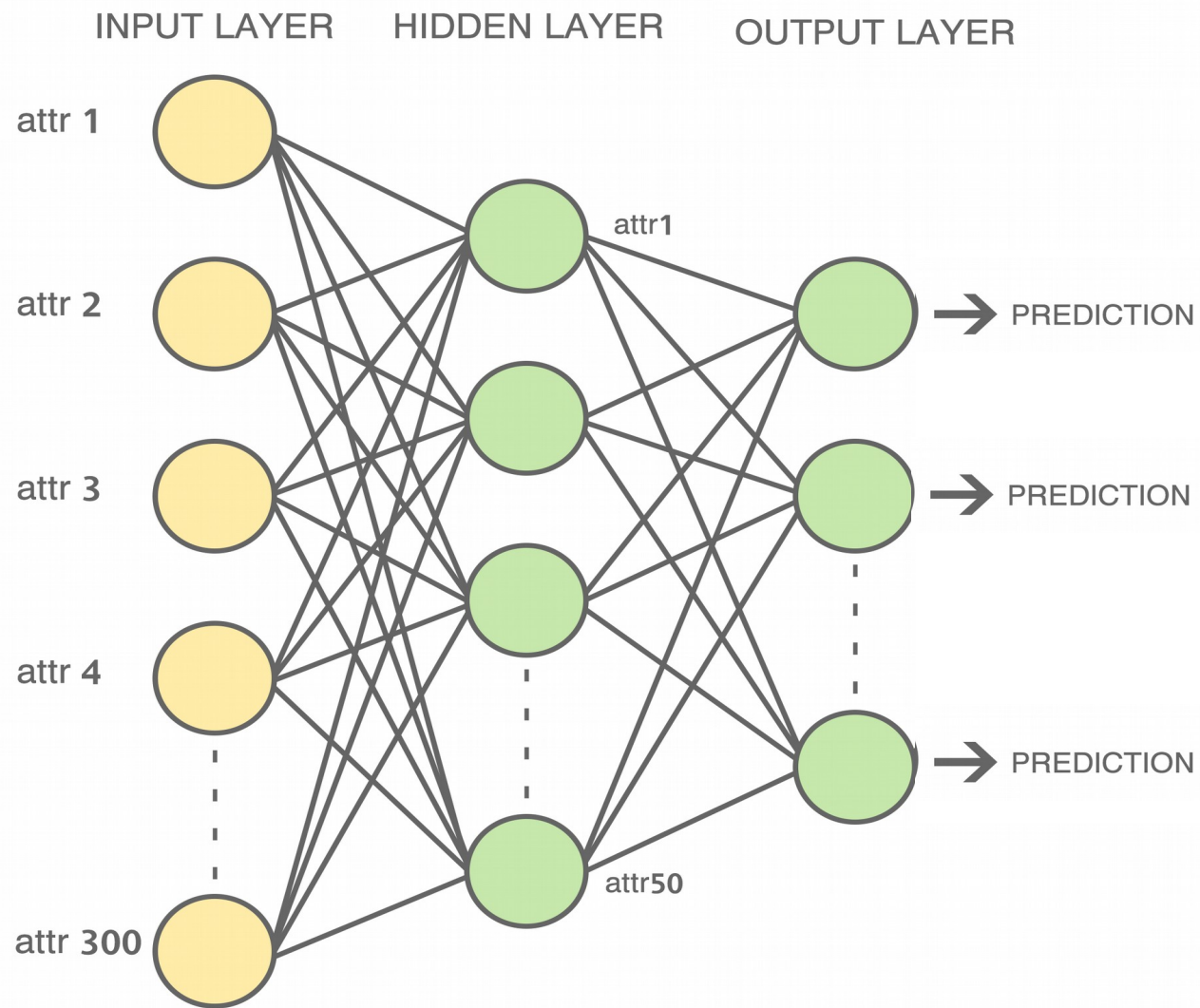
$\text{attr}_n$  Attribues

$h(\text{attr})$  Prédiction

# Deep Learning



# Deep Learning - Neural Networks



- Bibliothèque open source pour implémentation des algorithmes Machine Learning et Deep Learning.
- Créé par l'équipe « Google Brain » en 2011.
- S'appelait à l'origine « DistBelief ».
- Conçu pour le langage de programmation Python.
- Un API est disponible aussi pour C++.

- API de Haut Niveau, qui marche sur Tensorflow (et d'autre librairie)
- Créé pour assurer la simplicité et la facilité de l'implémentation des algorithmes ML/DL
- Intégré par GOOGLE dans Tensorflow.

# Installation Tensorflow/Keras

## 1 - Il faut installer PYTHON :

*Windows :* <https://www.python.org/downloads/>

*Linux :*

```
sudo apt-get install python3
```

## 2- Installer Tensorflow :

```
[ sudo ] pip install tensorflow
```



# Implémentation

## 1 - Créer un script python « train.py »



## 2 - Importer TensorFlow

```
import tensorflow as tf
```

## 3 - Importer les données x\_train, y\_train .. etc

att1	att2	att3	att4	att5	att6	at	classe
0.2468981	0.28536075	0.24272238	0.25871804	0.25487193	0.2530573	0.	4
0.25191244	0.26030546	0.2620809	0.25859925	0.25737765	0.23498312	0.	0
0.28840154	0.2417236	0.23060837	0.2246321	0.22908069	0.2449075	0.	4
0.24181724	0.26595387	0.27989298	0.27937967	0.2740329	0.2657349	0.	1
0.24482343	0.23654388	0.22716504	0.2244163	0.23051491	0.24280667	0.	0
0.25336483	0.2379957	0.24202825	0.22101313	0.21639593	0.21481487	0.	3
0.24170135	0.22695008	0.24261276	0.2511493	0.2568283	0.2534474	0.	3
0.27846226	0.25843102	0.26497927	0.25753054	0.2578077	0.24904782	0.	3

# Implémentation

## 4 - Conception du Modèle :

**1 - Construire un réseau simple et entièrement connecté :**

```
model = tf.keras.models.Sequential()
```

**2 - Ajouter les Couches (Layers) du réseau (Neural Network) :**

**COUCHE ENTRÉES (INPUT LAYER)**

```
model.add(tf.keras.layers.Dense(300, activation = tf.nn.relu))
```

**COUCHES CACHÉES (HIDDEN LAYERS)**

```
model.add(tf.keras.layers.Dense(50, activation = tf.nn.relu))
```

**COUCHE SORTIE (OUTPUT LAYERS)**

```
model.add(tf.keras.layers.Dense(5, activation = tf.nn.softmax))
```

## Fonction d'activation

comme l'hypothèse, mais son job est de calculer des nouveaux paramètres.

Exemples :

- **Soft-max** :

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

- **Sigmoïde** :

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

## 4 - Compilation (Création) du Modèle :

```
model.compile(  
    optimizer = 'adam',  
    loss = 'sparse_categorical_crossentropy',  
    metrics=['accuracy']  
)
```

- **Optimizer** : l'algorithme d'optimisation.
- **Loss** : Loss or Cost Function, la moyenne du tableau de sortie sur tous les points de données, autrement dit 'degré erreur'.
- **Metrics** : les métriques de mesure de qualité du modèle.

## 5 - Apprentissage (Learning) :

```
model.fit(x_train, y_train, epochs = 3)
```

### Epochs:

- (entier) Nombre d'époques pour former le modèle.
- Une époque est une itération sur l'ensemble des données x et y fournies

# Implémentation

## 6 - Évaluation du Modèle :

```
val_loss, val_acc = model.evaluate(x_test, y_test)
```

### - Autres Fonctions :

**Sauvegarde des poids :**

```
model.save_weights('weights.h5')
```

**Importation des poids :**

```
model.load_weights('weights.h5')
```

# Information

## Resources :

<https://www.deeplearning.ai> - Andrew NG Coursera

Grokking Deep Learning (Book)

Deep Learning With Python (Book)

## Code :

[https://github.com/logx7/tf\\_keras\\_prj](https://github.com/logx7/tf_keras_prj)

## Contact :

M.Benkedadra : [hammicristo@gmail.com](mailto:hammicristo@gmail.com)

Y.Youcefi : [yanilacamora@gmail.com](mailto:yanilacamora@gmail.com)

M.A.Benkorreche : [amine.benk27@gmail.com](mailto:amine.benk27@gmail.com)