



# Développeur d'Intelligence Artificielle Appliquée

## *Cours #9*

[www.impactia.org](http://www.impactia.org)

# Structure de la formation

- **#1: Introduction**
- **#2: Vision**
- **#3: Vision**
- **#4: Vision**
- **#5: Renforcement**
- **#6: Renforcement**
- **#7: Renforcement**
- **#8: Langage**
- **#9: Langage**
- #10: Langage
- #11: Génération d'images
- #12: Génération d'images
- #13: Génération d'images
- #14: Projet
- #15: Projet

# Semaine dernière : Cours #8

- Théorie
  - Fin de la partie Renforcement
  - Début de la partie Langage : rapide historique
- Pratique
  - Entrainement d'agents droneRL
  - Challenge droneRL ! 

# Aujourd'hui : Cours #9

- Pratique
  - Génération de texte avec PyTorch

# Traitement du Langage Naturel (NLP)

# Comment entraîner un modèle ?

**2010-2017: entraînement de zéro**  
Création d'un nouveau modèle pour chaque tâche

**2018: modèles pré-entraînés**  
Réutilisation de modèles grâce à l'apprentissage par transfert



# Comment entraîner un modèle ?

## 2019: apprentissage “few shots”

Création d'un nouveau modèle pour chaque tâche



## 2020: apprentissage “zero shot”

Réutilisation de modèles grâce à l'apprentissage par transfert

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



# Comment entraîner un modèle ?

**2019: apprentissage “few shots”**  
Création d'un nouveau modèle pour chaque tâche

**2020: apprentissage “zero shot”**  
Réutilisation de modèles grâce à l'apprentissage par transfert

"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile** [I'm not a fool].

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose,**" which translates as, "Lie lie and something will always remain."

"I hate the word '**perfume**'," Burr says. 'It's somewhat better in French: '**parfum**'.

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre côté? -Quel autre côté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

**"Brevet Sans Garantie Du Gouvernement"**, translated to English: "**Patented without government warranty**".

*Table 1.* Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

# Comment entraîner un modèle ?

## 2021: interface conversationnel

Interaction avec le modèle sous forme d'une discussion

## 2021 : apprentissage "RLHF"

Amélioration du modèle en utilisant du renforcement de l'apprentissage par les retours humains



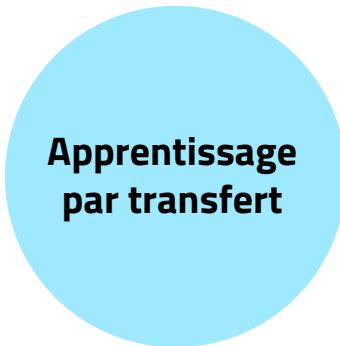
ChatGPT 4

You  
Can you translate this to French? "plush girafe"

ChatGPT  
The phrase "plush giraffe" translates to French as "girafe en peluche."

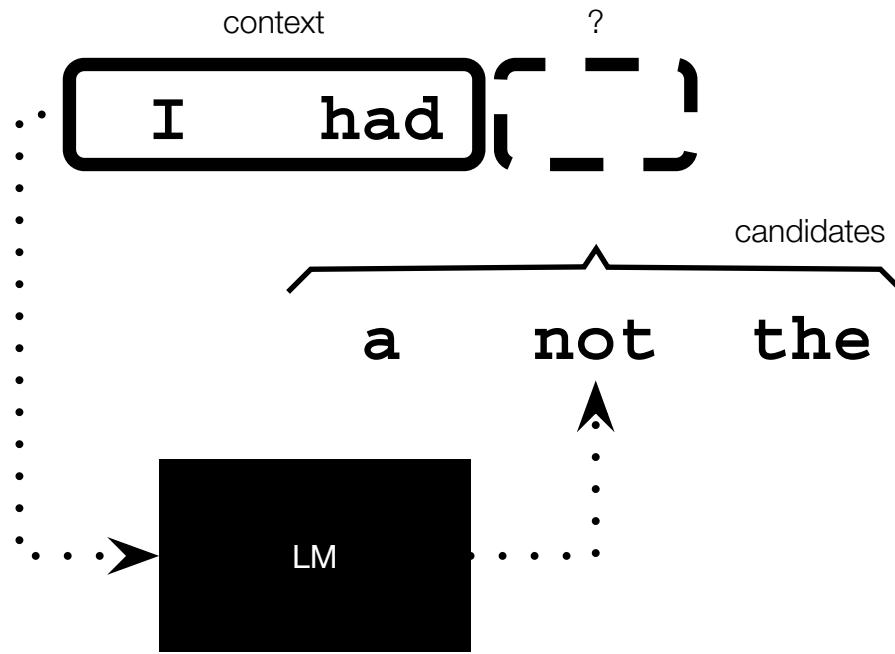
Speaker icon, file icon, circular arrow icon, thumbs down icon

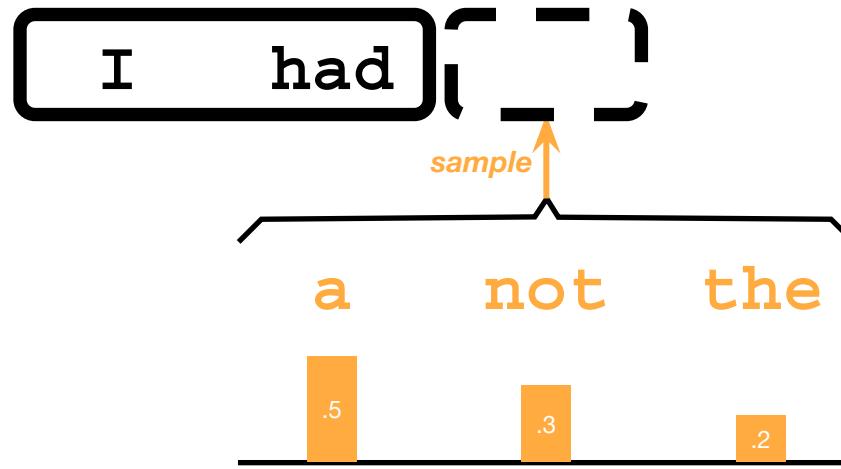
# Les approches de formations



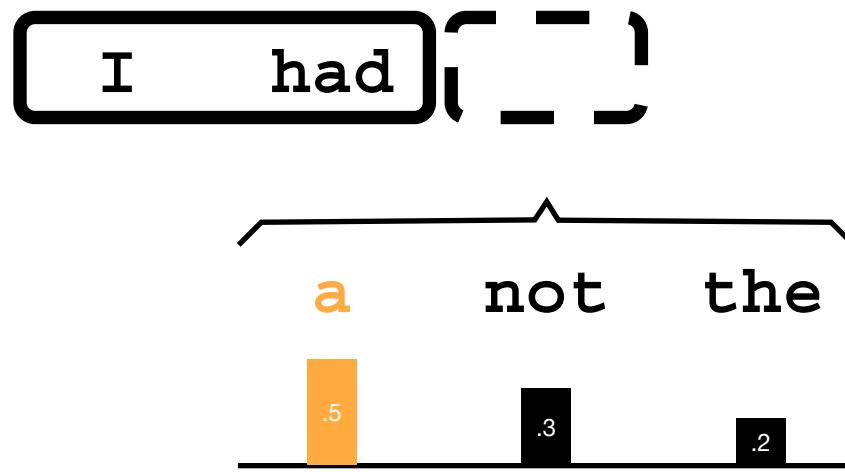
I had [ - ]

<https://openai.com/blog/better-language-models/>

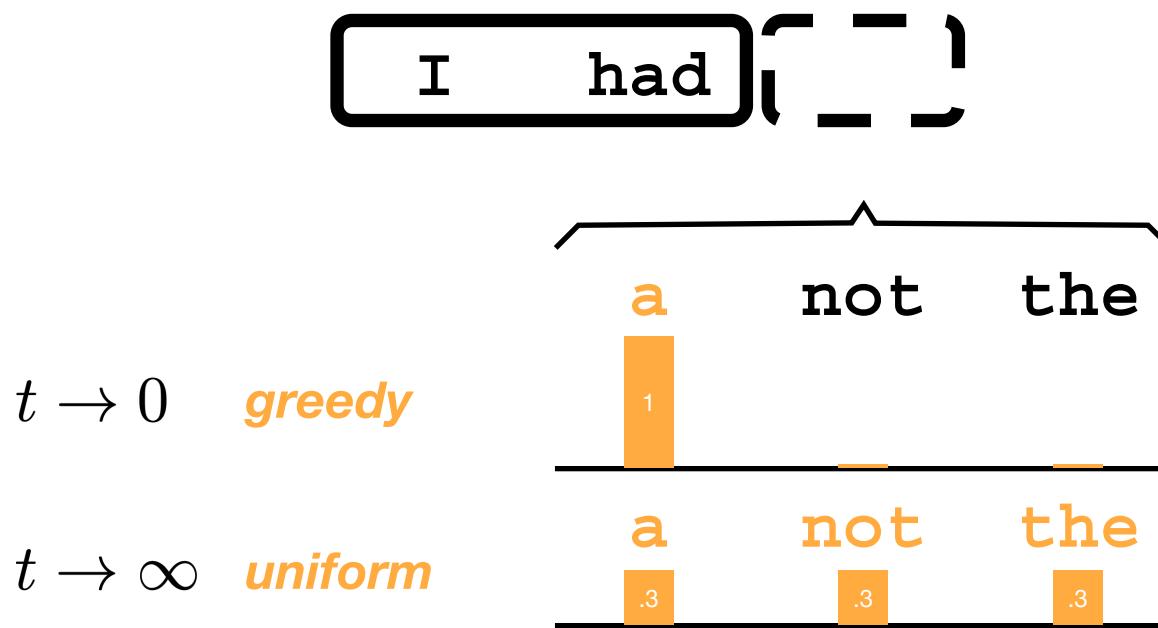




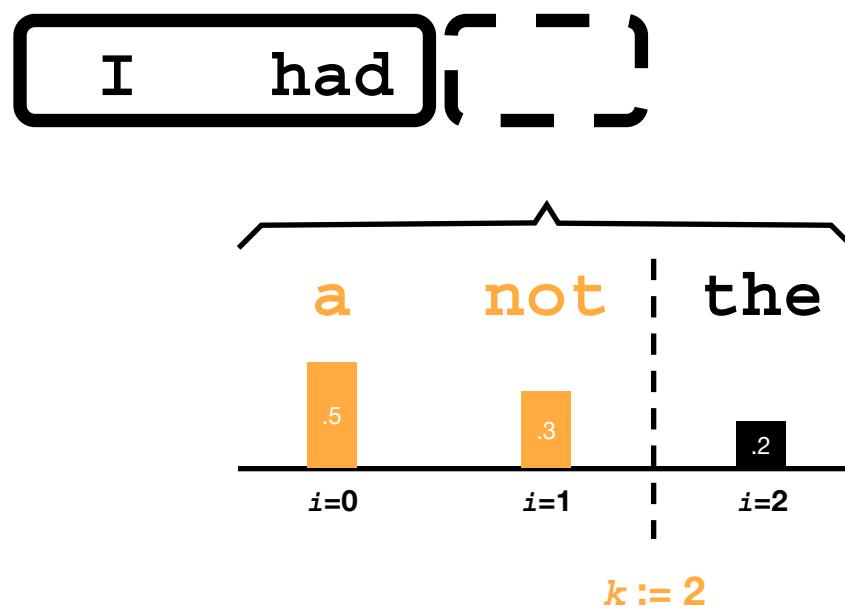
# greedy



# temperature $t$



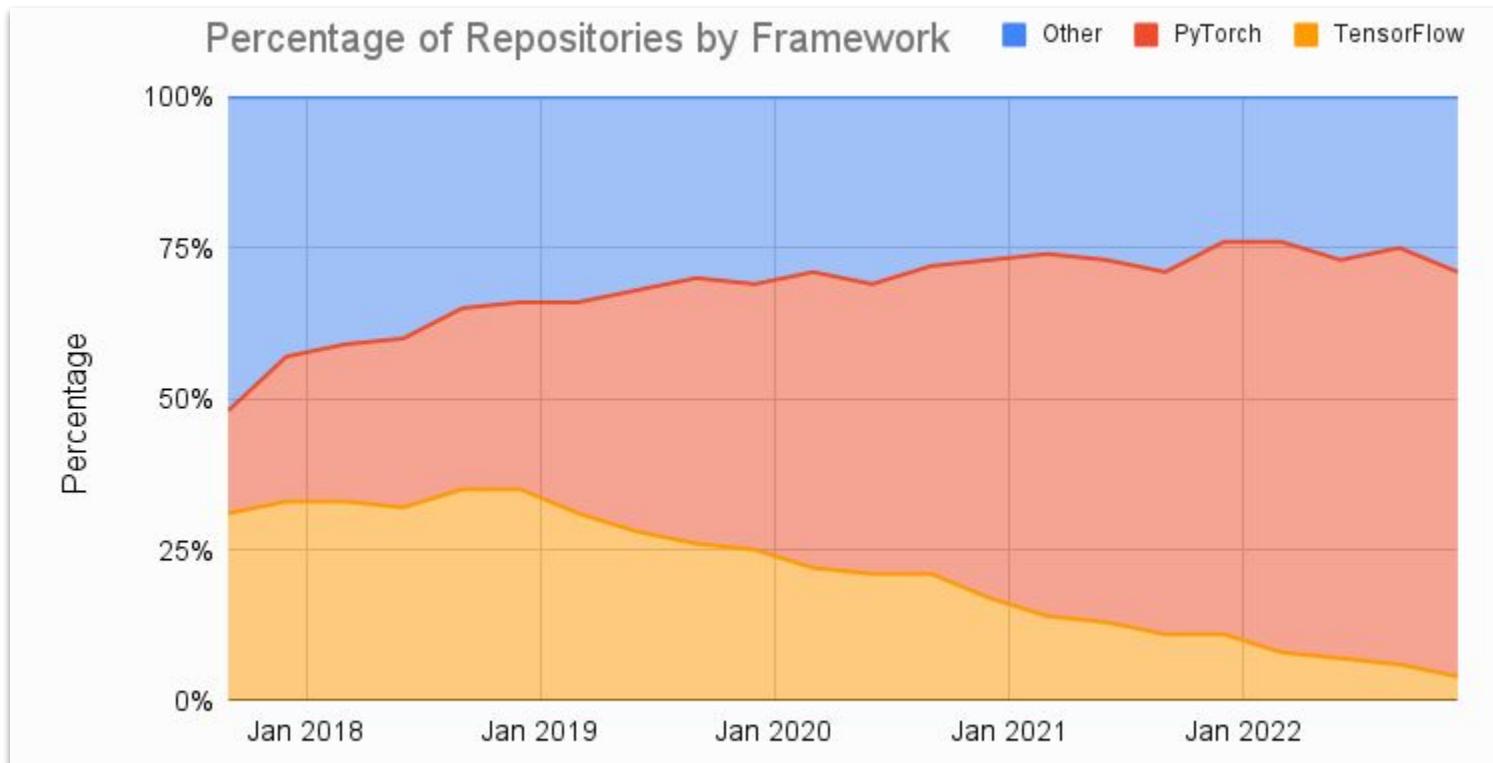
# top- $k$



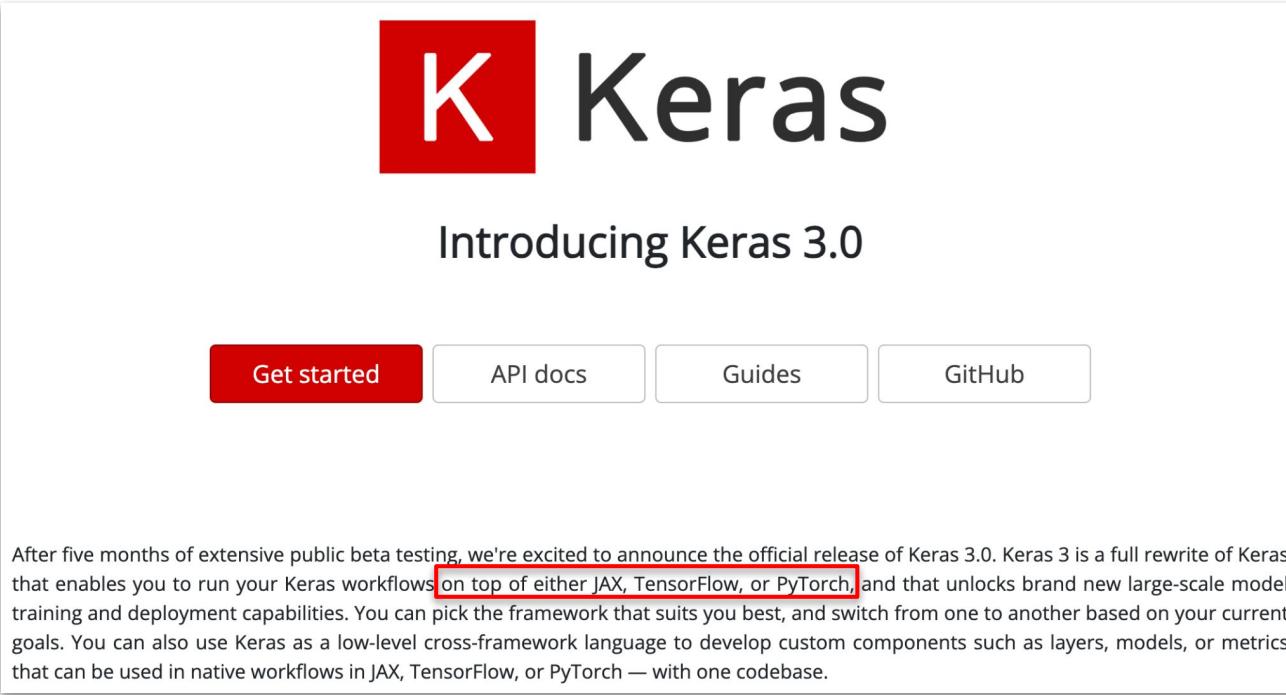
# Génération de texte en pratique

# Librairies de Machine Learning

TensorFlow  
vs PyTorch



# Librairies de Machine Learning



The screenshot shows the official landing page for Keras 3.0. At the top is a large red square icon containing a white stylized letter 'K'. To the right of the icon, the word 'Keras' is written in a large, bold, black sans-serif font. Below this title, the text 'Introducing Keras 3.0' is centered in a smaller, dark gray sans-serif font. Underneath the title, there is a horizontal navigation bar with four items: 'Get started' (in a red button), 'API docs', 'Guides', and 'GitHub'. A paragraph of text below the navigation bar discusses the release of Keras 3.0, mentioning its rewrite, support for JAX, TensorFlow, or PyTorch, and new model training capabilities. The text is presented in a dark gray sans-serif font.

After five months of extensive public beta testing, we're excited to announce the official release of Keras 3.0. Keras 3 is a full rewrite of Keras that enables you to run your Keras workflows on top of either JAX, TensorFlow, or PyTorch, and that unlocks brand new large-scale model training and deployment capabilities. You can pick the framework that suits you best, and switch from one to another based on your current goals. You can also use Keras as a low-level cross-framework language to develop custom components such as layers, models, or metrics that can be used in native workflows in JAX, TensorFlow, or PyTorch — with one codebase.

[https://keras.io/keras\\_3](https://keras.io/keras_3)

Novembre 2023

# Reconnaissance d'images avec Keras

```
[ ] from keras import layers

model = keras.Sequential(
    [
        layers.Flatten(input_shape=(28, 28)),
        layers.Dense(200, activation="relu"),
        layers.Dense(100, activation="relu"),
        layers.Dense(75, activation="relu"),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

[ ] 28*28*200

[ ] # "résumé" de notre modèle
model.summary()

[ ] model.compile(
    loss="categorical_crossentropy",
    optimizer="adam",
    metrics=["accuracy"]
)

[ ] model.fit(
    x_train, y_train,
    batch_size=128,
    epochs=3,
    validation_split=0.1
)

[ ] model.evaluate(x_test, y_test, verbose=0)
```

# Apprentissage par renforcement avec droneRL

```
# Création des agents
agents = {drone.index: RandomAgent(env) for drone in env.drones}
agents[0] = DQNAgent(
    env,
    ConvQNetworkFactory(
        env,
        conv_layers=[
            {'out_channels': 32, 'kernel_size': 3, 'stride': 1, 'padding': 1},
            {'out_channels': 32, 'kernel_size': 3, 'stride': 1, 'padding': 1},
            {'out_channels': 64, 'kernel_size': 3, 'stride': 1, 'padding': 1},
            {'out_channels': 64, 'kernel_size': 3, 'stride': 1, 'padding': 1},
        ],
        dense_layers=[
            1024, 256
        ],
        gamma=0.95,
        epsilon_start=1,
        epsilon_decay=0.99,
        epsilon_end=0.01,
        memory_size=10000,
        batch_size=64,
        target_update_interval=5
    )
    trainer = MultiAgentTrainer(env, agents, reset_agents=True, seed=0)
    agents[0].qnetwork

# Entrainement
# on appelle trainer.train plusieurs fois pour changer d'environnement
for run in range(4):
    trainer.train(1000)
```

# Génération de texte avec PyTorch

```
In [795...]:  
for i in range(200000):  
  
    # minibatch construct  
    ix = torch.randint(0, Xtr.shape[0], (32,))  
  
    # forward pass  
    emb = C[Xtr[ix]] # (32, 3, 10)  
    h = torch.tanh(emb.view(-1, 30) @ W1 + b1) # (32, 200)  
    logits = h @ W2 + b2 # (32, 27)  
    loss = F.cross_entropy(logits, Ytr[ix])  
    #print(loss.item())  
  
    # backward pass  
    for p in parameters:  
        p.grad = None  
    loss.backward()  
  
    # update  
    #lr = lrs[i]  
    lr = 0.1 if i < 100000 else 0.01  
    for p in parameters:  
        p.data += -lr * p.grad  
  
    # track stats  
    #lri.append(lre[i])  
    stepi.append(i)  
    lossi.append(loss.log10().item())  
  
#print(loss.item())
```

Mise en pratique

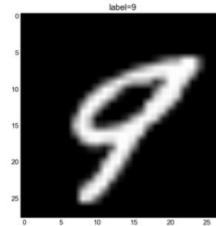
## Génération de texte

<https://colab.research.google.com/#create=true>



# Encodage des labels

Image :



Label :

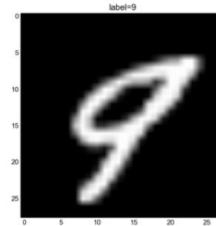
9

Label,  
représenté en "one-hot" :

0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1

# Encodage des labels

Image :



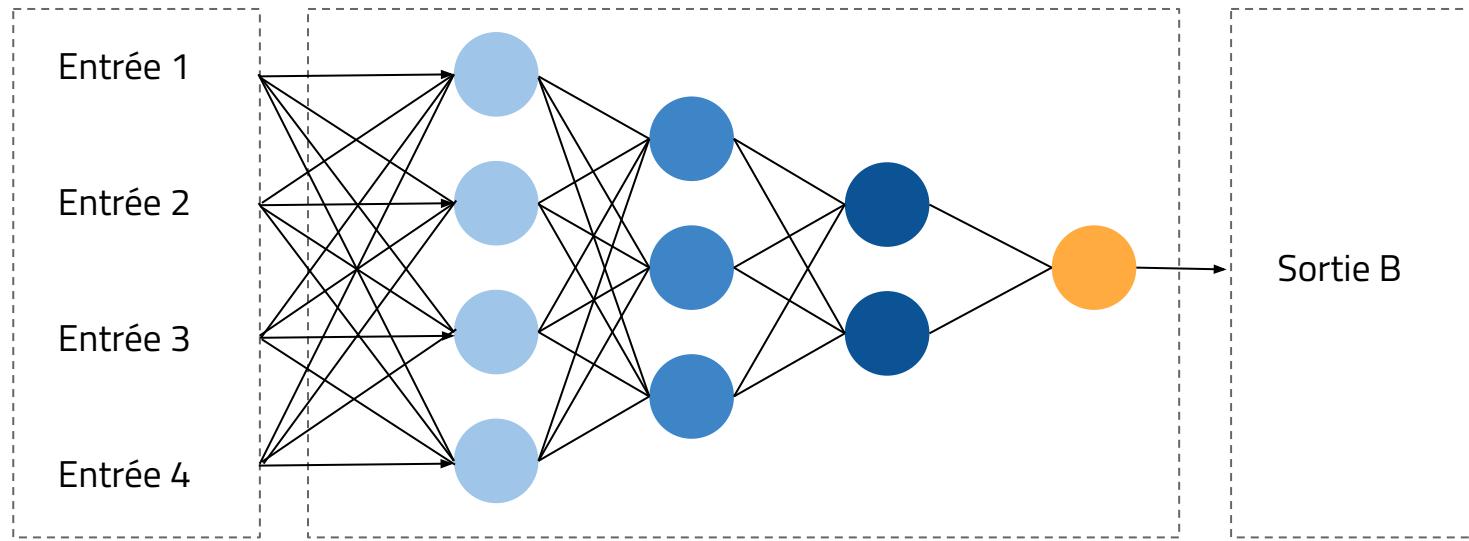
Label :

9

Label,  
représenté en "one-hot" :

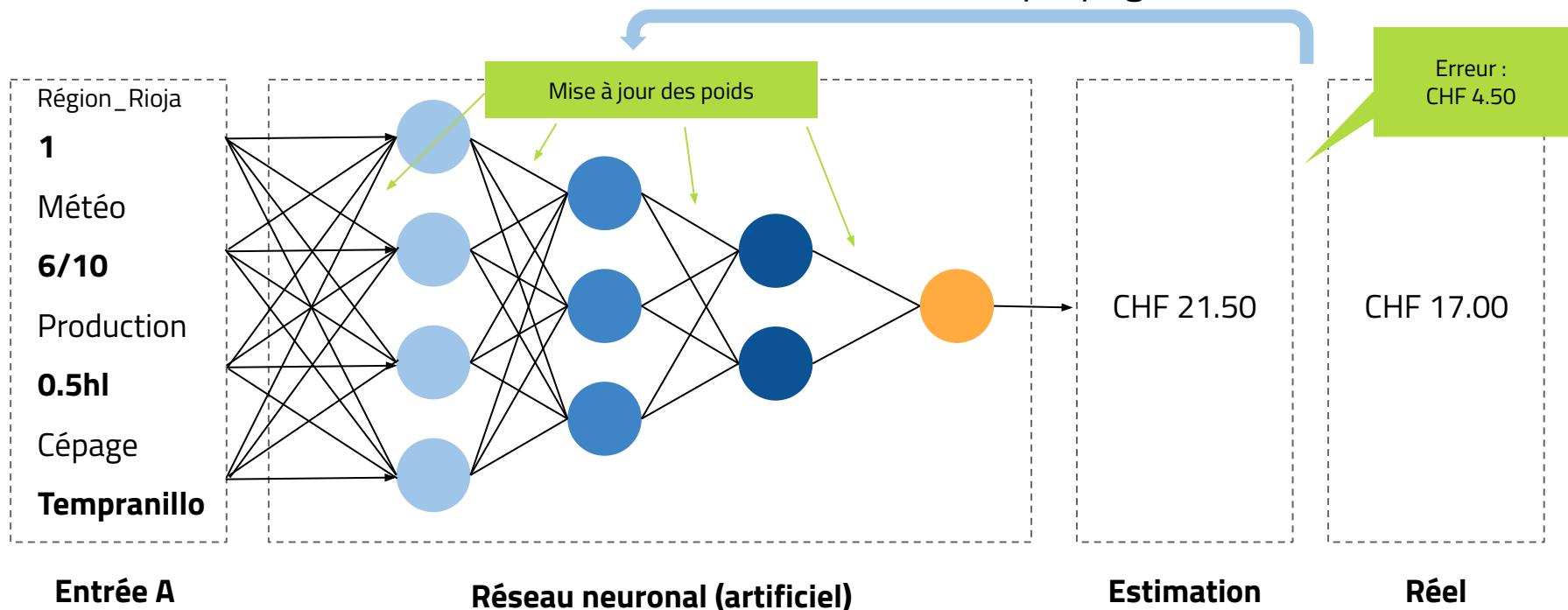
0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1

# Réseaux neuronaux

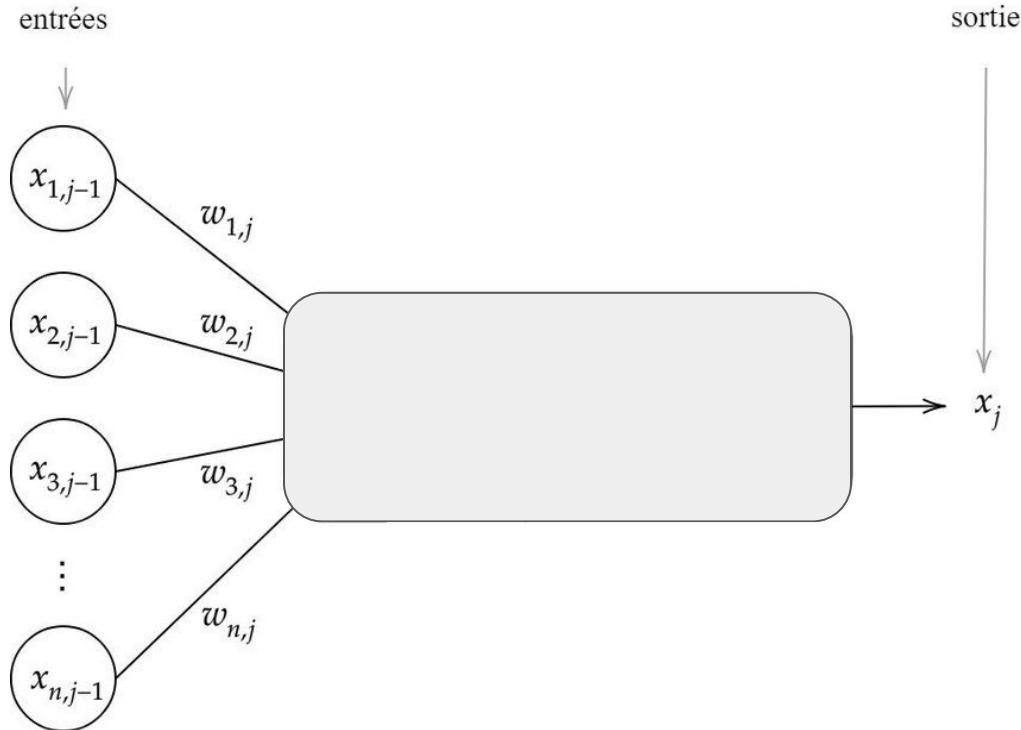


**Modèle**

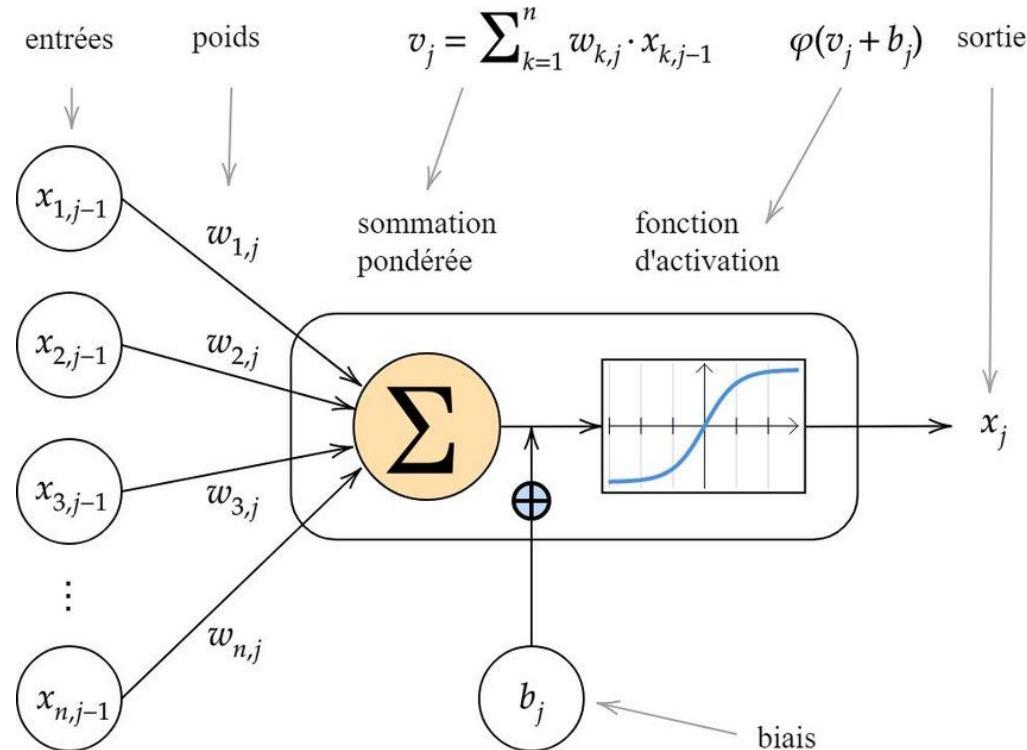
# Entrainement supervisé



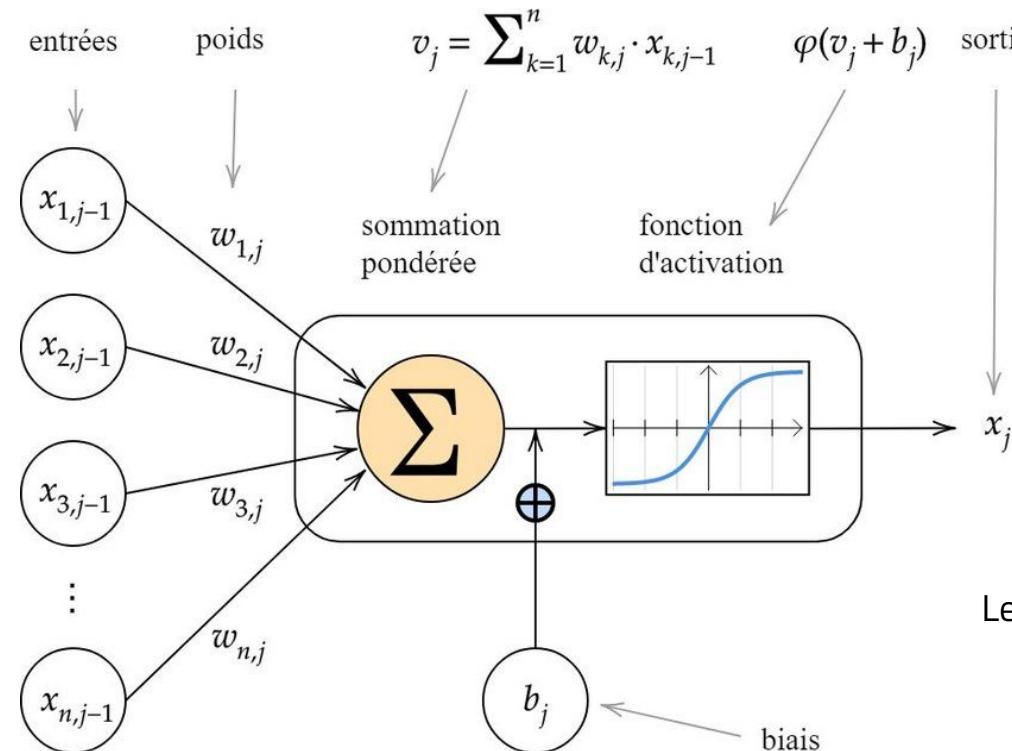
# Regardons un seul neurone



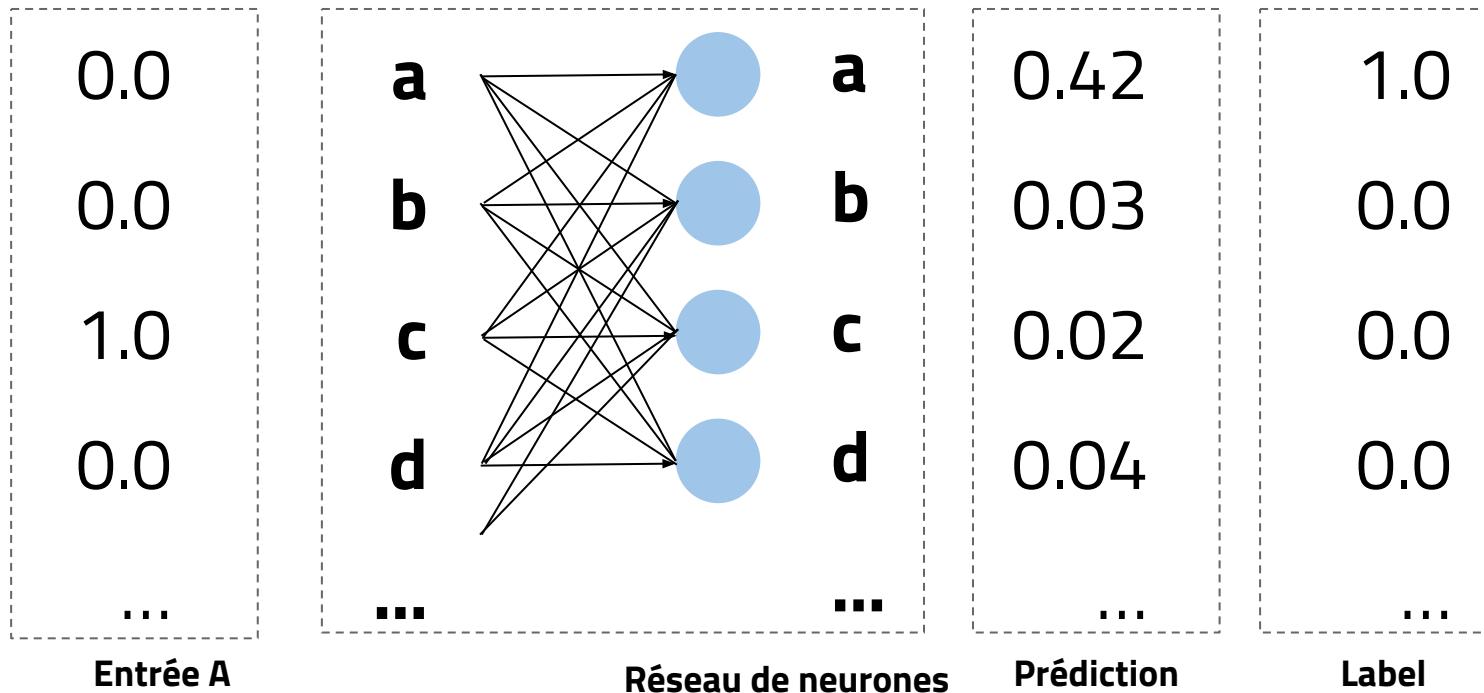
# Regardons un seul neurone



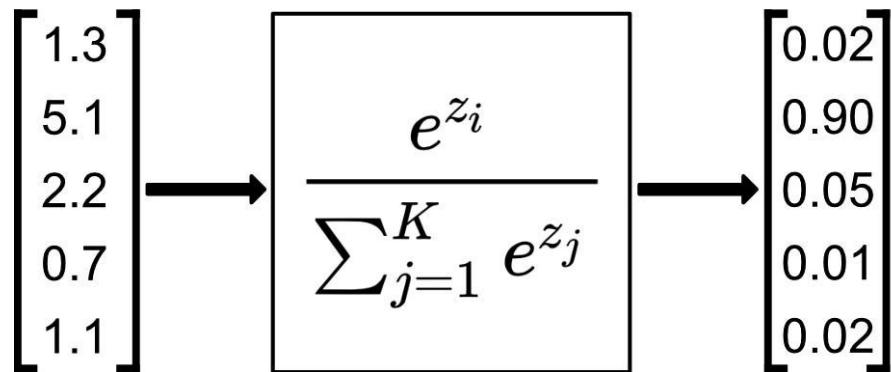
# Regardons un seul neurone



# Entrainement supervisé pour classification



# Fonction Softmax pour la Classification



# Entraînement supervisé pour classification

