

Fiche élève - Introduction au Deep Learning

Solutions detaillées avec explications :

1. Surapprentissage (Overfitting) :

- Definition :

Le surapprentissage survient lorsque le modèle apprend trop précisément sur les données d'entraînement,

y compris les bruits ou anomalies, au lieu de généraliser pour de nouvelles données.

- Risques :

Cela mène à des performances élevées sur l'entraînement mais faibles sur les données de test ou réelles.

- Identifier le surapprentissage :

- Ecart important entre précision d'entraînement et précision de validation/test.

- Exemple : Précision d'entraînement 0.99 mais test seulement 0.85.

2. Notion de bruit :

- Definition :

Une convergence bruitée signifie que la précision ou la perte fluctue significativement entre les lots.

- Pourquoi cela se produit :

Les lots plus petits introduisent des variations dues à la petite taille d'échantillon utilisée pour mettre à jour les poids.

- Impact :

Cela peut ralentir la convergence mais parfois aide à éviter les minima locaux.

Fiche élève - Introduction au Deep Learning

3. Difference entre les fonctions d activation :

- Sigmoid :

- Sortie : entre 0 et 1, interpretee comme une probabilite.
- Problemes : Saturation (gradient nul pour les grandes valeurs).

- Tanh :

- Sortie : entre -1 et 1, meilleure pour les donnees centrees autour de 0.
- Avantage : Plus de robustesse face aux gradients nuls.

- Exemple pratique :

tanh est souvent utilise pour les couches cachees, sigmoid pour la derniere couche pour des probabilites.

4. Simulations avec explications :

- Variation des epoques :

- Code :

```
model.fit(x_train, y_train, epochs=5, batch_size=32, validation_split=0.2)
```

```
model.fit(x_train, y_train, epochs=10, batch_size=32, validation_split=0.2)
```

- Explication :

Avec 5 epoques, le modele peut ne pas avoir assez de temps pour apprendre (sous-apprentissage).

Avec 10 epoques, l apprentissage est meilleur mais risque de surapprentissage si validation ne s ameliore pas.

- Modification de la taille des lots :

- Code :

Fiche élève - Introduction au Deep Learning

```
model.fit(x_train, y_train, epochs=5, batch_size=16, validation_split=0.2)
```

```
model.fit(x_train, y_train, epochs=5, batch_size=64, validation_split=0.2)
```

- Explication :

Avec `batch_size=16` : Mise a jour des poids plus frequente, plus de bruit mais convergence rapide.

Avec `batch_size=64` : Convergence plus lente, mais plus stable grace a un apprentissage regulier.

- Fonction d activation :

- Code :

```
model = Sequential([  
    Flatten(input_shape=(28, 28)),  
    Dense(128, activation='tanh'),  
    Dense(10, activation='softmax')  
])
```

- Explication :

Avec `tanh` : utile pour une propagation symetrique autour de 0, adapte a des couches intermediaires.

Avec `sigmoid` : excellent pour les probabilites mais attention au probleme de saturation.

En resume, ajuster les hyperparametres (nombre d epoques, taille des lots, fonctions d activation) est essentiel pour equilibrer precision, vitesse et capacite de generalisation du modele.