

LE POLYTECHNIQUE DE L’UNIVERSITE FRANÇOIS RABELAIS DE TOURS

Spécialité Informatique

64 avenue Jean Portalis,

37200 TOURS, FRANCE

Tél +33 (0)2 47 36 14 14

[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr/)

Rapport - TP2

FENG Jiaming 21707371

SONG Miyuan 21707400

## 

### Sklearn

Scikit-learn (sklearn) est un module tiers couramment utilisé dans le machine learning. Il encapsule les méthodes d'apprentissage automatique couramment utilisées, y compris la régression, la réduction de dimension, la classification et le clustering. Méthode. Lorsque nous sommes confrontés à un problème d'apprentissage automatique, nous pouvons choisir la méthode correspondante selon la figure suivante. Sklearn présente les caractéristiques suivantes:

Des outils simples et efficaces d'exploration de données et d'analyse de données permettent à chacun de réutiliser NumPy, Scipy et MatPlotLib dans des environnements complexes.

### Pytorch

PyTorch est un package Python basé sur la bibliothèque Torch, conçu pour accélérer les applications d'apprentissage en profondeur.

PyTorch fournit une méthode abstraite similaire à NumPy pour caractériser les tenseurs (ou tableaux multidimensionnels, les scalaires sont des tenseurs d'ordre zéro, les vecteurs sont des tenseurs du premier ordre et les matrices sont des tenseurs du second ordre). Il peut utiliser des GPU pour accélérer la formation. La torche peut remplacer Numpy dans les réseaux de neurones.

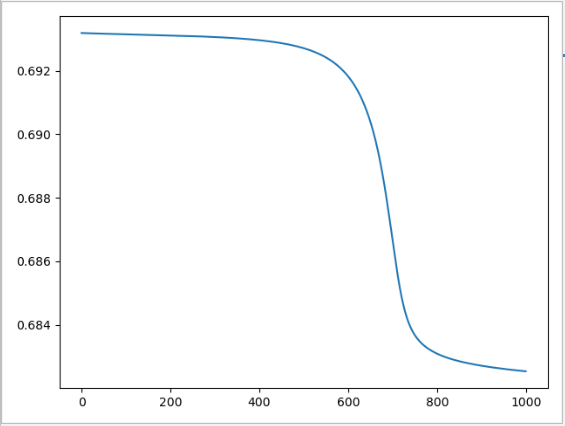
Lorsque vous utilisez pytorch pour créer des réseaux de neurones, il existe deux packages les plus couramment utilisés: **torch.nn** et **torch.optim**. Le package torch.nn contient principalement des modules pour la construction de différentes couches, telles que la convolution bidimensionnelle entièrement connectée, la mise en commun, etc.; le package torch.nn contient également une série de fonctions de perte utiles. Ces fonctions sont également Indispensable lors de l'entraînement de réseaux de neurones, tels que CrossEntropyLoss, MSELoss, etc.

Le package torch.optim contient principalement des algorithmes d'optimisation utilisés pour mettre à jour des paramètres, tels que SGD, AdaGrad, RMSProp, Adam, etc.

Il existe deux méthodes couramment utilisées pour définir le réseau à l'aide du package torch.nn. L'une consiste à hériter de la classe **nn.Module**. Cette méthode peut personnaliser la structure du réseau. Surtout lorsqu'il est nécessaire d'implémenter des paramètres partagés, nous pouvons simplement La fonction directe utilise à plusieurs reprises la même couche définie dans la fonction \_\_init\_\_; l'autre est définie à l'aide de **torch.nn.Sequential**

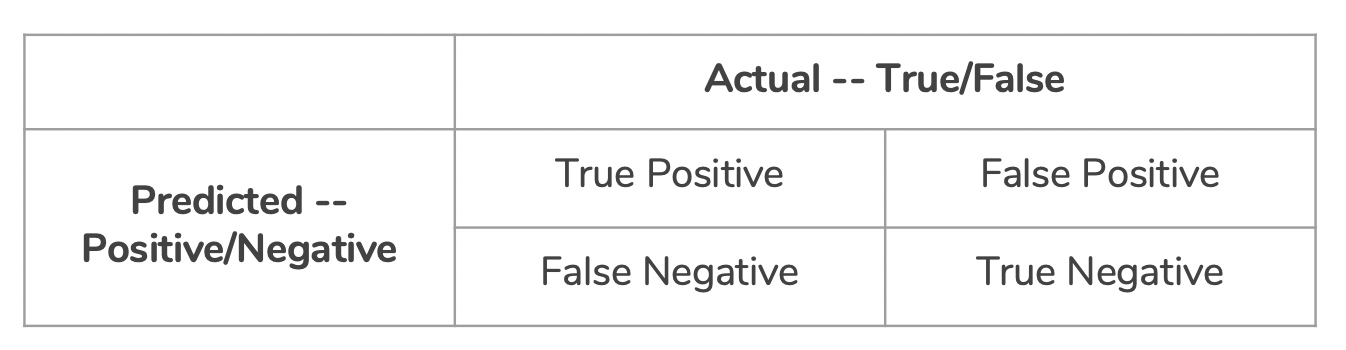
1. **Visualizer l'évolution de la loss, tracez la courbe de perte.**

1000 itérations de changements de pertes moyennes sont indiquées ci-dessous：



**2.Calculer la précision du modèle sur la base d'apprentissage.**

La classification est une tâche relativement courante dans le machine learning. Pour les indicateurs d'évaluation communs aux tâches de classification, il y a **l’accuracy** et la **précision**.

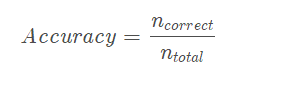


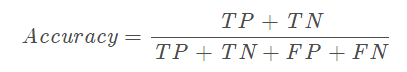
Comme le montre la figure ci-dessus, pour comprendre chaque indice d'évaluation, vous devez d'abord connaître la matrice de confusion. P dans la matrice de confusion est positif, et N est négatif.

Dans le tableau, FP indique le nombre d'échantillons qui sont réellement négatifs mais qui devraient être positifs, TN indique le nombre d'échantillons qui sont réellement négatifs et qui devraient être négatifs, TP indique le nombre d'échantillons qui sont réellement positifs et qui devraient être positifs, et FN indique que le nombre est réellement positif mais est prévu pour être positif. Le nombre d'échantillons devrait être négatif.

De plus, TP + FP représente le nombre de tous les échantillons qui devraient être positifs. De même, FN + TN est le nombre de tous les échantillons qui devraient être négatifs, TP + FN est le nombre d'échantillons qui sont réellement positifs et FP + TN est le nombre d'échantillons qui sont réellement négatifs.

### L’accuracy

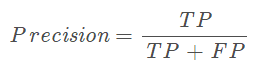
Le taux de accuracy est la proportion d'échantillons correctement classés par rapport au nombre total d'échantillons, c'est-à-dire:

En combinant la matrice de confusion ci-dessus, la formule peut également être écrite comme ceci:

L’accuracy est l'indice d'évaluation le plus simple et le plus intuitif des problèmes de classification, mais elle présente des défauts évidents. Par exemple, si 99% des échantillons sont des échantillons positifs, le classificateur n'a besoin que de prédire positivement tout le temps pour obtenir une accuracy de 99%, mais ses performances réelles sont très faibles. En d'autres termes, lorsque la proportion d'échantillons dans différentes catégories est très inégale, la catégorie avec une grande proportion devient souvent le facteur le plus important affectant l’accuracy .

### La précision

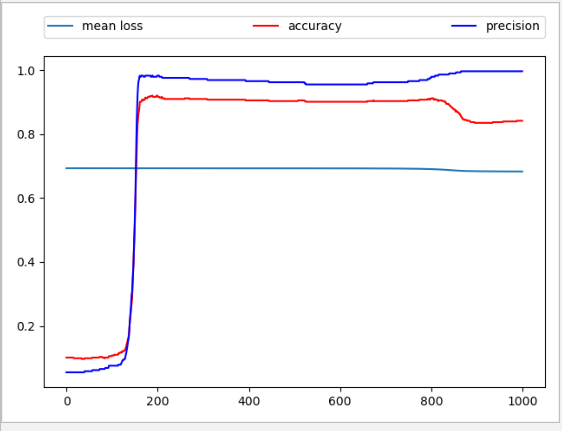
La précision fait référence à la proportion d'échantillons qui sont réellement positifs parmi les échantillons que le modèle prévoit comme positifs. La formule de calcul est:



Dans cette partie, nous utilisons les deux fonctions suivantes pour calculer la précision.



Le résultat est montré dans la figure



**3.Créer une base de validation et une base de test**