



Jak ocenić sprawność modelu?



Jak ocenić sprawność modelu?

Aby poprawnie określić, czy model dobrze nauczył się przewidywać to, do czego jest stworzony, musimy zastosować statystyki pomiędzy prawdą a przewidywaniami modelu. Do tego celu używa się F1-score, dokładności i precyzji.

Statystyki

Precyzja

F1-score

Dokładność

F1-score

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

F1-score

F1-score jest jedną z najprostszych statystyk które można zastosować w celu sprawdzenia jak dobrze model nauczył się przewidywania. Aby ją przeprowadzić, musimy określić liczbę wyników prawdziwie pozytywnych (TP), liczbę wyników fałszywie pozytywnych (FP) i liczbę wyników fałszywie negatywnych (FN). Im bliżej wartość statystyki jest równa 1, tym lepiej.

Dokładność

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Dokładność

Dokładność jest pomiarem ile przewidywań jest zgodnych z prawdą w skali procentowej. Liczy się ją sumując wyniki prawdziwie pozytywne (TP) z wynikami fałszywie pozytywnymi (FP) i dzieląc przez całość zbioru przewidywanego.

Precyzja

$$\textit{Precision} = \frac{TP}{TP + FP}$$

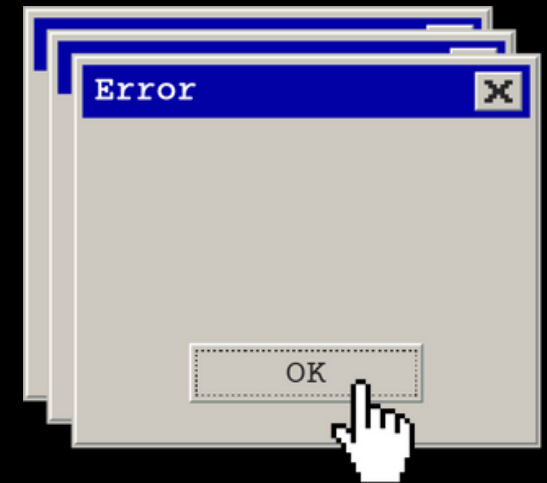
Precyzja

Precyzja mierzy dokładność prawdziwych pozytywnych przewidywań modelu w stosunku do sumy prawdziwych i fałszywych przewidywań modelu. Im bliżej ta wartość jest 1, tym lepiej model poprawnie przewiduje wyniki i unika generacji wyników fałszywie pozytywnych.



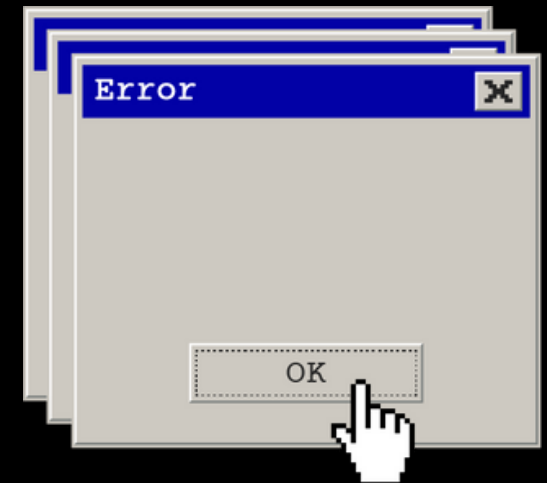
Dopasowanie modelu do danych

- **MSE** (Mean Squared Error)
- **MAE** (Mean Absolute Error)
- **RMSE** (Root Mean Squared Error)
- **R2** (Współczynnik determinancji)



Dopasowanie modelu do danych

- **MSE** (Mean Squared Error)
- **MAE** (Mean Absolute Error)
- **RMSE** (Root Mean Squared Error)
- **R2** (Współczynnik determinancji)



MSE

- średnia kwadratowa błędów
- im niższa wartość, tym lepsze jest dopasowanie modelu do danych

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

MAE

- średni błąd bezwzględny
- bardziej odporna na dane odstające
- im mniejszy tym lepsze dopasowanie

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

RMSE

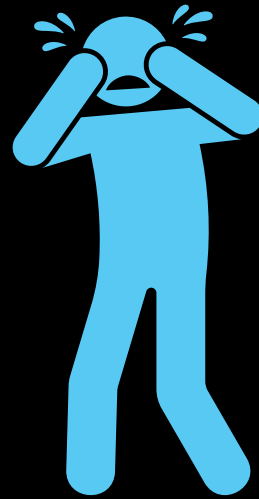
- pierwiastek kwadratowy z MSE
- jest w tych samych jednostkach co dane, co ułatwia interpretację
- im niższy tym lepsze dopasowanie

R²

- proporcja wariancji w zmiennej zależnej, która jest przewidywana przez zmienne niezależne
- jak dobrze model wyjaśnia zmienność danych
- im bliżej 1, tym dopasowanie modelu jest lepsze

$$R^2 := \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \geq 0$$

Funkcje straty w trenowaniu



wyliczają błąd między przewidywaniami a prawdą i na jego podstawie optymalizują model

Funkcje straty w trenowaniu

MODELE
REGRESYJNE

MSELoss (MSE)

L1Loss (MAE)

MODELE KLASYFIKACJI
WIELOKLASOWEJ I BINARNEJ

CrossEntropyLoss

BCELoss

MODELE
OPARTE O SVM

HingeLoss

Optymalizatory modelu

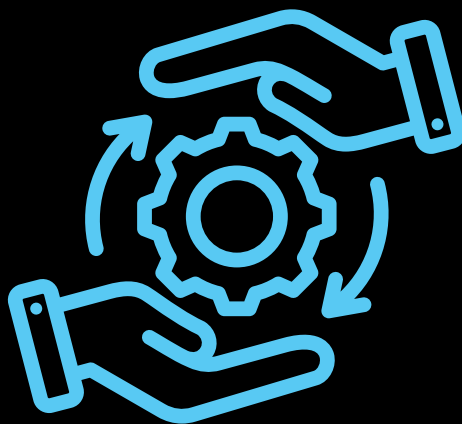
Optymalizatory służą do aktualizacji wag modelu w trakcie procesu uczenia. Ich głównym celem jest minimalizacja funkcji straty, co pozwala na poprawę dokładności modelu. W PyTorch dostępnych jest wiele różnych optymalizatorów, z których każdy ma swoje unikalne właściwości i zastosowania.

Optymalizatory modelu:

Główne funkcje optymalizatorów



Aktualizacja wag



Zarządzanie
współczynnikiem uczenia



Obsługa momentu

np.: Adam, SGD

Adam

Adam to jeden z najczęściej używanych optymalizatorów, który łączy zalety dwóch innych metod: AdaGrad i RMSProp. Używa adaptacyjnych współczynników uczenia dla różnych parametrów. Oblicza średnie momenty pierwszego i drugiego rzędu dla gradientów, co pozwala na dostosowanie współczynnika uczenia w czasie.

SGD

SGD to klasyczny algorytm optymalizacji, który aktualizuje wagi na podstawie losowo wybranych próbek z danych treningowych. W każdej iteracji SGD oblicza gradient na podstawie pojedynczej próbki (lub małej partii) i aktualizuje wagi w kierunku przeciwnym do gradientu.

