

Models de l'Optimització en l'Aprenentatge Automàtic (Machine Learning)

Dra Sundus Zafar

Introducció

Objectiu: Entendre com els mètodes d'optimització són utilitzats en l'aprenentatge automàtic per entrenar ajustar models, entrenar hiperparàmetres, i millorar el rendiment.

Aplicacions:

- ▶ Optimització de funcions de pèrdua.
- ▶ Ajust d'hiperparàmetres (per exemple, Gradient Descent, Grid Search, Random Search).
- ▶ Regularització per evitar sobreajustaments.

Problema General d'Optimització

Donat un conjunt de paràmetres $\theta \in \Theta$, trobem:

$$\theta^* = \arg \min_{\theta \in \Theta} f(\theta)$$

On:

- ▶ $f(\theta)$: Funció objectiu (p. ex., funció de pèrdua).
- ▶ Θ : Espai de cerca (hiperparàmetres possibles).

Exemple: Minimitzar l'error quadràtic mitjà (MSE):

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

Optimització en Aprenentatge Automàtic

En l'aprenentatge automàtic, el procés d'optimització és fonamental per:

- ▶ Augmentar el rendiment del model.
- ▶ Minimitzar la funció de pèrdua (error).

Els problemes d'optimització es poden classificar en dos tipus principals:

- ▶ **Optimització per augmentar el rendiment:** Com aconseguir que el model funcioni millor.
- ▶ **Optimització per reduir la pèrdua:** Com reduir l'error o la pèrdua en el model.

Optimització per Augmentar el Rendiment

Objectiu: Trobar els paràmetres del model que maximitzen la seva eficiència.

Exemple: Optimització de la precisió d'un model de classificació.

$$\text{Rendiment} = \text{Accuracy} = \frac{\text{Verdader Positiu} + \text{Verdader Negatiu}}{\text{Total Observacions}}$$

En aquest cas, volem augmentar la precisió del model ajustant els seus paràmetres d'entrenament (com la taxa d'aprenentatge, profunditat de l'arbre, etc.).

Optimització per Augmentar el Rendiment

Problema d'Optimització: Maximitzar la funció de rendiment subjecta a restriccions:

$$\text{Maximitzar } f(\theta) \quad \text{subjecte a } g(\theta) \leq 0$$

on $f(\theta)$ és una mesura de rendiment (per exemple, l'accuracy), i $g(\theta)$ és una restricció del model (per exemple, la complexitat).

Optimització per Reduir la Pèrdua

Objectiu: Minimitzar l'error del model per millorar les seves prediccions.

Exemple: Optimització de la funció de pèrdua en la regressió lineal.

$$\text{Minimitzar: } L(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

En aquest cas, la funció de pèrdua $L(\theta)$ mesura la diferència entre les prediccions \hat{y}_i i els valors reals y_i .

Optimització per Reduir la Pèrdua

Problema d'Optimització: Minimitzar la funció de pèrdua per ajustar els paràmetres θ :

$$\min_{\theta} L(\theta) \quad \text{on } \theta \in \mathbb{R}^d$$

on θ són els paràmetres del model, i $L(\theta)$ és la funció de pèrdua (per exemple, l'error quadràtic mitjà).

Exemple d'Optimització per Reduir la Pèrdua (Regressió Lineal)

Exemple: Minimització de la pèrdua en regressió lineal mitjançant Gradient Descent.

Funció de pèrdua:

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

$$\hat{y}_i = X_i \cdot \theta$$

on X_i és el vector de característiques per a la i -èsima observació.

Comparació d'Optimització per Rendiment i Pèrdua

Diferències Clau:

- ▶ **Optimització per augmentar el rendiment:** Ens centra en obtenir el millor rendiment del model (per exemple, màxima precisió, recall, etc.).
- ▶ **Optimització per reduir la pèrdua:** Ens centra en ajustar els paràmetres del model per minimitzar l'error o la discrepància entre les prediccions i els valors reals.

Exemple:

- ▶ **Rendiment:** Millorar la precisió de classificació.
- ▶ **Pèrdua:** Minimitzar l'error quadràtic mitjà (MSE) en regressió.

Gradient Descent

Idea: Iterativament actualitzem els paràmetres en la direcció negativa del gradient per minimitzar la funció objectiu.

Actualització:

$$\theta^{t+1} = \theta^t - \eta \nabla f(\theta^t)$$

On:

- ▶ η : Taxa d'aprenentatge.
- ▶ $\nabla f(\theta^t)$: Gradient de f a θ^t .

Exemple: Entrenament d'una regressió logística.

Grid Search

Definició: Cerca exhaustiva d'hiperparàmetres en una graella predefinida.

Procediment:

1. Definir una graella Θ_{grid} amb combinacions d'hiperparàmetres.
2. Per cada $\theta \in \Theta_{\text{grid}}$, calcular $f(\theta)$.
3. Escollir el millor:

$$\theta^* = \arg \min_{\theta \in \Theta_{\text{grid}}} f(\theta)$$

Exemple:

- ▶ Model: SVM.
- ▶ Hiperparàmetres: $C \in \{0.1, 1, 10\}$, kernel $\in \{\text{lineal}, \text{RBF}\}$.

Random Search

Definició: Cerca d'hiperparàmetres mitjançant mostreig aleatori.

Procediment:

1. Definir l'espai de cerca Θ .
2. Seleccionar k mostres aleatòries de Θ .
3. Escollir el millor:

$$\theta^* = \arg \min_{\theta \in \{\theta_1, \dots, \theta_k\}} f(\theta)$$

Exemple:

- ▶ Model: Random Forest.
- ▶ Hiperparàmetres: Profunditat màxima, nombre d'arbres.

Bayesian Optimization

Definició: Modelització probabilística de $f(\theta)$ com un procés gaussià (GP).

Procediment:

1. Inicialitzar amb avaluacions aleatòries.
2. Modelitzar $f(\theta)$ amb un GP.
3. Utilitzar una funció d'adquisició (p. ex., Expected Improvement):

$$\theta_{\text{next}} = \arg \max_{\theta \in \Theta} a(\theta)$$

4. Actualitzar el model i repetir.

Exemple: Hiperparàmetres d'una xarxa neuronal (nombre de capes, taxa d'aprenentatge).

Conclusions

- ▶ L'optimització és central per l'aprenentatge automàtic.
- ▶ Diversos mètodes com Gradient Descent, Grid Search, Random Search, i Bayesian Optimization són essencials per ajustar models.
- ▶ La selecció del mètode depèn dels recursos computacionals i la complexitat del model.
- ▶ Cada algorisme d'optimització té avantatges i desavantatges.
- ▶ Gradient Descent és eficaç per entrenar models, mentre que Grid i Random Search són útils per ajustar hiperparàmetres.
- ▶ Bayesian Optimization és ideal per problemes costosos amb espais de cerca grans.

- ▶ La optimització en aprenentatge automàtic és clau per millorar el rendiment dels models.
- ▶ Els problemes d'optimització per reduir la pèrdua i per augmentar el rendiment tenen aproximacions similars però objectius diferents.
- ▶ Els algorismes d'optimització com Gradient Descent, Grid Search, Random Search, i Bayesian Optimization s'utilitzen per resoldre aquests problemes.

Passos següents: Experimentar amb els algorismes en diferents datasets.