DL : L10 : 19.05.2022

6.2 Weight norm penalty

Change on the cost function:

old cost function : $L(\underline{\theta})$ based on KL divergence

New regularized cost f^n:

$$L_r(\underline{\theta}) = L(\underline{\theta}) + \sum_{l=1}^{L} \lambda_l P(\underline{\underline{W}}_l)$$

$P(\underline{\underline{W}}_l) \geq 0$: <u>Penalty terms</u> ; Penalize $\underline{\theta}$ with large $P(\underline{\underline{W}}_l)$

$\lambda_l \geq 0$: <u>Regularization parameters</u>

→ A compromise between $\min L(\underline{\theta})$ and $\min P(\underline{\underline{W}}_l)$

→ no unique sol^n as a multiobjective, conflicting problems.

$\lambda_l = 0 \; \forall l$: no regularization

λ_l is high → then $L(\underline{\theta})$ gets impacted.

<u>Common choice of $P(\underline{\underline{W}}_l)$</u>

(a) l2 regularization : use L2 norm of vec $(\underline{\underline{W}})$

$$P(\underline{\underline{W}}_l) = \|\text{vec}(\underline{\underline{W}})\|_2^2 = \sum_j \sum_i W_{l,ij}^2 \mathrel{\hat{=}} \text{weight energy}$$

→ prefer $\underline{\theta}$ with small weight energy

<u>Analysis</u>:

$L_r(\underline{\theta}) = L(\underline{\theta}) + \lambda \|\theta\|^2$ for simplicity.

$$\nabla L_r(\underline{\theta}) = \nabla L(\underline{\theta}) + 2\lambda \underline{\theta}$$

L2 regularization leads to weight decay during training

$$\underline{\theta}^{t+1} = \underline{\theta}^t - \gamma^t \cdot \nabla L_r(\underline{\theta})$$

try to lower the original weight...

$$= \underline{\theta}^t - \gamma^t \nabla L(\underline{\theta}) - 2\lambda \gamma^t \underline{\theta}^t = \underbrace{(1 - 2\lambda\gamma^t)}\underline{\theta}^t - \gamma^t \nabla L(\underline{\theta})$$

(b) L1 - regularization : use $l1$ norm of $\text{vec}(\underline{\underline{W}}_l)$

$$\rho(\underline{\underline{W}}_l) = \|\text{vec}(\underline{\underline{W}}_l)\|_1 = \sum\sum |w_{l,ij}|$$

→ prefer sparse $\underline{\underline{W}}_l$ with many zero elements.

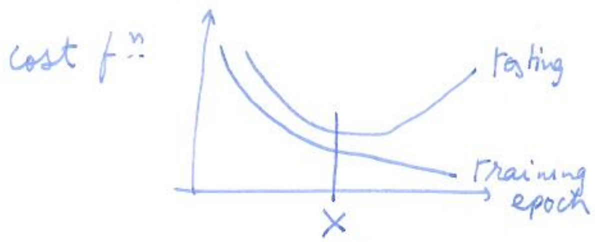
Bias $\underline{b}_l$:

- no amplification of the input vector $\underline{x}_{l-1}$
- no need for regularization

Slide 6-5

6.3 Early stopping

Change on optimizer



Stop the training. Here there is a divergence in the training error rate and test error rate

Slide 6-6

6.4 Data augmentation

change on dataset

Overfitting - more complex model which can memorize the training dataset. Theoretically infinite no of training data will never overfit any neural network

But we have limited dataset.

Data augmentation - Generate artificial but realistic training ~~angles~~ data / samples.

6.5 Ensemble learning

Change on dataset / maodel / cost function / optimizer

Slide 6-8

6.6 Dropout

Change in model

An implicit ensemble learning method

Slide 6-9, 6-10

Co-adapted — ?

6.7 Hyperparameters optimization

6-17, 18, 19
Slide

(a) Can't do many hyperparameter optimization as they are mostly integer optimization. or discrete value optimisation

(b) Can be solved by test set or validation set

dataset D

previously D_{train} D_{test}

now D_{train} D_{test} D_{valid}

D_{train} ↓ training set ↓ training the model

D_{test} ↓ test set

D_{valid} ↓ Validation test ↓ tuning the hypermeters

Early stopping is a type of hyperparameter optimization on no of epochs.

Training and hyperparameter optimization

for $\underline{\eta} = \ldots$

- learn $\underline{\theta}$ of $f(\underline{x}; \underline{\theta}, \underline{\eta})$ from D_{train}
- Calculate validation error $(\underline{\eta})$ of $f(\underline{x}; \underline{\theta}, \underline{\eta})$ on D_{val}

end

$\min\limits_{\underline{\eta}}$ validation error $(\underline{\eta})$

Calculate test error of $f(\underline{x}; \underline{\theta}; \underline{\eta})$ on D_{test}

If $err_{test} > err_{train}$ or err_{val} then use more well-defined optimization techniques.

Slide 6-20

Q Any mathematical approach for Bayesian optimisation?
Ilias