Deep Learning

Advanced Topics

Machine Learning

Dongsuk Yook
Artificial Intelligence Laboratory
Korea University

Contents

Class Objectives

- Understanding some advanced methods of deep learning to improve the performance
- Being able to apply the advanced techniques when building deep neural networks

Contents

- Weight Update Schedules
- Momentum
- Learning Rates
- Normalization
- Regularization
- Hyperparameters

Gradient Descent

- Gradient descent
  - $\theta \leftarrow \theta - \eta \nabla E$

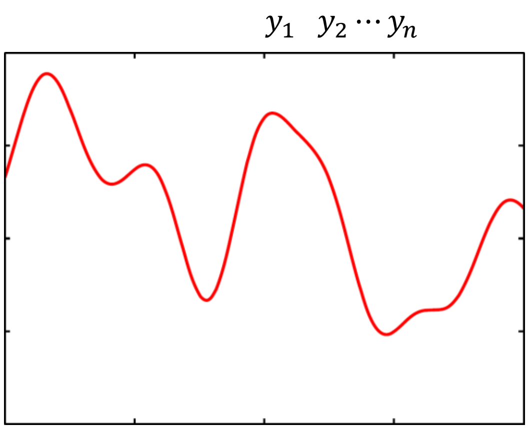
- Weight update schedules
  - Batch training
  - Stochastic training
  - Online training
  - Learning with queries

Batch Gradient Descent

- Input: $\mathcal{D} = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_{t=1}^N$

Initialize $\{\mathbf{W}_l\}_{l=1}^L$ with small random numbers.
repeat ; epoch
  $\Delta\mathbf{W}_l \leftarrow \mathbf{0}$ for all l
  for each $(\boldsymbol{x}^t, \boldsymbol{r}^t) \in \mathcal{D}$
    for each layer index l from 1 to L
      $\boldsymbol{z}_l \leftarrow \boldsymbol{a}_l(\mathbf{W}_l \boldsymbol{z}_{l-1})$ } $z_{l,i} \leftarrow 1/\left(1+\exp\left(-\boldsymbol{w}_{l,i}^{\top}\boldsymbol{z}_{l-1}\right)\right)$ for all i
    end
    $\boldsymbol{\delta}_L \leftarrow \nabla_{\boldsymbol{s}_L}\mathcal{L}(\boldsymbol{r}^t, \boldsymbol{y})$ } $\delta_{L,i} \leftarrow r_i^t - y_i$ for all i ; $\boldsymbol{y} \equiv \boldsymbol{z}_L$, $\boldsymbol{s}_L \equiv \mathbf{W}_L \boldsymbol{z}_{L-1}$
    for each layer index l from L to 1
      $\Delta\mathbf{W}_l \leftarrow \Delta\mathbf{W}_l + \eta\boldsymbol{\delta}_l \boldsymbol{z}_{l-1}^{\top}$ } $\Delta w_{l,ij} \leftarrow \Delta w_{l,ij} + \eta\delta_{l,i} z_{l-1,j}$ for all i and j
      if $(1 < l)$ $\boldsymbol{\delta}_{l-1} \leftarrow \mathbf{W}_l^{\top}\boldsymbol{\delta}_l \odot \boldsymbol{z}_{l-1}'$ } $\delta_{l-1,j} \leftarrow \sum_i \delta_{l,i} w_{l,ij}\, z_{l-1,j}'$ for all j
    end
  end
  for each layer index l from 1 to L
    $\mathbf{W}_l \leftarrow \mathbf{W}_l + \Delta\mathbf{W}_l$; weight update per epoch
  end
until convergence

- Output: $\{\mathbf{W}_l\}_{l=1}^L$
- Parallelization
- Similar or sometimes redundant samples
- Local minima



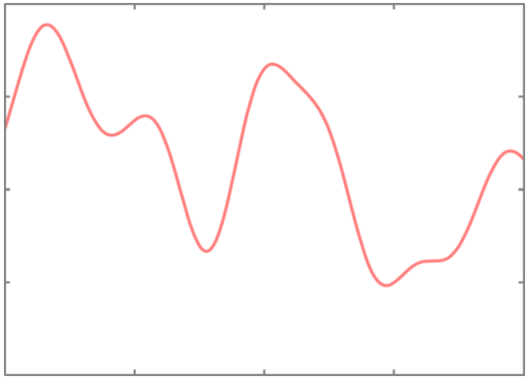
Stochastic Gradient Descent

- Input: $\mathcal{D} = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_{t=1}^N$

Initialize $\{\mathbf{W}_l\}_{l=1}^L$ with small random numbers.
repeat ; epoch
  for each $(\boldsymbol{x}^t, \boldsymbol{r}^t) \in \mathcal{D}$ in random order
    for each layer index l from 1 to L
      $\boldsymbol{z}_l \leftarrow \boldsymbol{a}_l(\mathbf{W}_l \boldsymbol{z}_{l-1})$; feed forward, $\boldsymbol{z}_0^t \equiv \boldsymbol{x}^t$
    end
    $\boldsymbol{\delta}_L \leftarrow \nabla_{\boldsymbol{s}_L} \mathcal{L}(\boldsymbol{r}^t, \boldsymbol{y})$; $\boldsymbol{y} \equiv \boldsymbol{z}_L$, $\boldsymbol{s}_L \equiv \mathbf{W}_L \boldsymbol{z}_{L-1}$
    for each layer index l from L to 1
      if $(1 < l)$ $\boldsymbol{\delta}_{l-1} \leftarrow \mathbf{W}_l^\top \boldsymbol{\delta}_l \odot \boldsymbol{z}_{l-1}'$; error back propagate
      $\mathbf{W}_l \leftarrow \mathbf{W}_l + \eta \boldsymbol{\delta}_l \boldsymbol{z}_{l-1}^\top$; weight update per sample
    end
  end
until convergence

- Output: $\{\mathbf{W}_l\}_{l=1}^L$

- Random walk
- Fluctuation around the minima
- Learning rate scheduling (simulated annealing)
- No parallelization



Mini-Batch Stochastic Gradient Descent

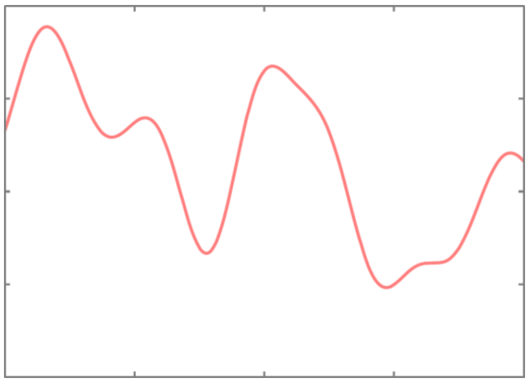
- Input: $\mathcal{D} = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_{t=1}^{N}$

Initialize $\{\mathbf{W}_l\}_{l=1}^{L}$ with small random numbers.
repeat ; epoch
    for each random mini-batch $\mathcal{m} \subset \mathcal{D}$
        $\Delta\mathbf{W}_l \leftarrow \mathbf{0}$ for all l
        for each $(\boldsymbol{x}^t, \boldsymbol{r}^t) \in \mathcal{m}$
            for each layer index l from 1 to L
                $\boldsymbol{z}_l \leftarrow \boldsymbol{a}_l(\mathbf{W}_l \boldsymbol{z}_{l-1})$; feed forward, $\boldsymbol{z}_0^t \equiv \boldsymbol{x}^t$
            end
            $\boldsymbol{\delta}_L \leftarrow \nabla_{\boldsymbol{s}_L} \mathcal{L}(\boldsymbol{r}^t, \boldsymbol{y})$; $\boldsymbol{y} \equiv \boldsymbol{z}_L$, $\boldsymbol{s}_L \equiv \mathbf{W}_L \boldsymbol{z}_{L-1}$
            for each layer index l from L to 1
                $\Delta\mathbf{W}_l \leftarrow \Delta\mathbf{W}_l + \eta \boldsymbol{\delta}_l \boldsymbol{z}_{l-1}^{\top}$; accumulate gradient
                if $(1 < l)$ $\boldsymbol{\delta}_{l-1} \leftarrow \mathbf{W}_l^{\top} \boldsymbol{\delta}_l \odot \boldsymbol{z}_{l-1}'$; error back propagate
            end
        end
        for each layer index l from 1 to L
            $\mathbf{W}_l \leftarrow \mathbf{W}_l + \Delta\mathbf{W}_l$; weight update per mini-batch
        end
    end
until convergence

- Output: $\{\mathbf{W}_l\}_{l=1}^{L}$

- Parallelization

- Mini-batch size

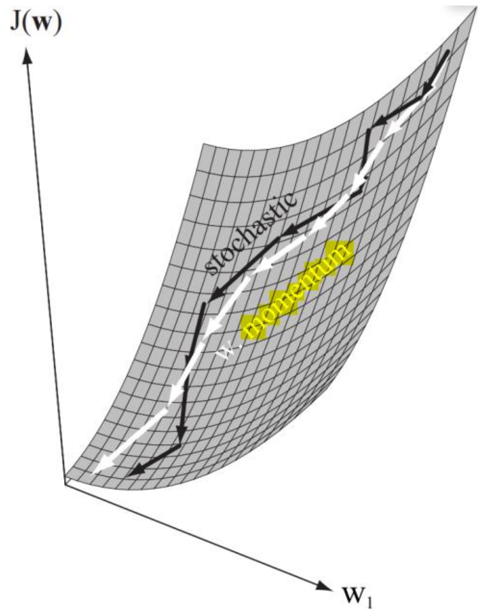


Contents

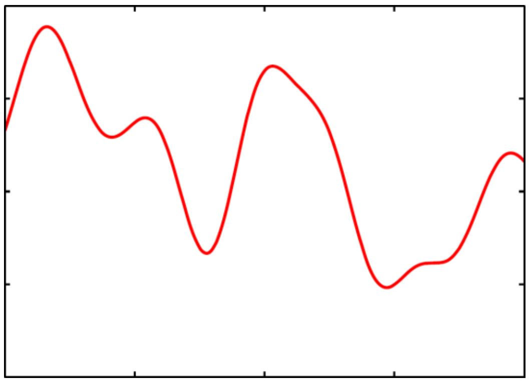
Momentum

- Gradient descent
  - $w^{t+1} \leftarrow w^t - \eta \frac{\partial E^t}{\partial w}$ $\quad ; \frac{\partial E^t}{\partial w} \equiv \left.\frac{\partial E}{\partial w}\right|_t$
  - $\Delta w^t \leftarrow -\eta \frac{\partial E^t}{\partial w}$
    $w^{t+1} \leftarrow w^t + \Delta w^t$

- Gradient descent with *momentum*
  - $\Delta w^t \leftarrow \alpha \Delta w^{t-1} - \eta \frac{\partial E^t}{\partial w} \Rightarrow \Delta w^t \leftarrow \alpha \Delta w^{t-1} - \underbrace{(1-\alpha)\acute{\eta}}_{\eta} \frac{\partial E^t}{\partial w}$
    $w^{t+1} \leftarrow w^t + \Delta w^t$

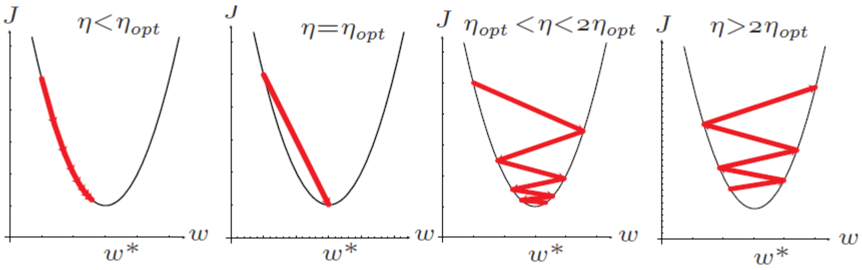


[Duda *et al*. 2001]

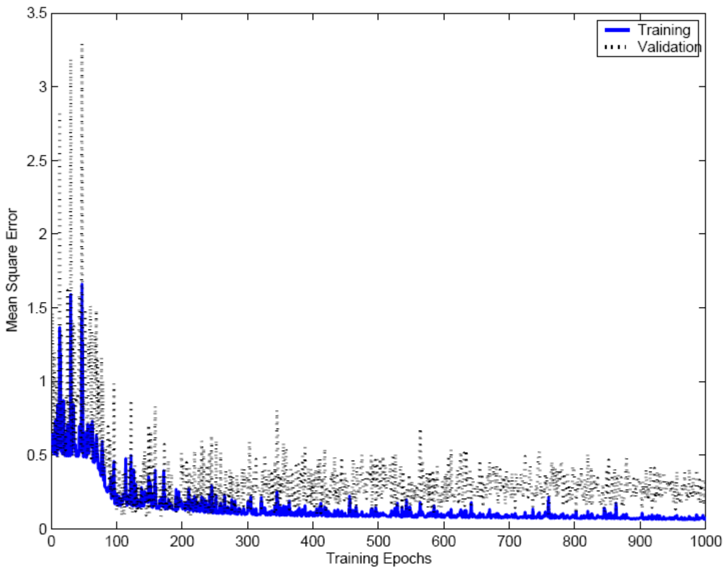


Contents

Learning Rate Decay



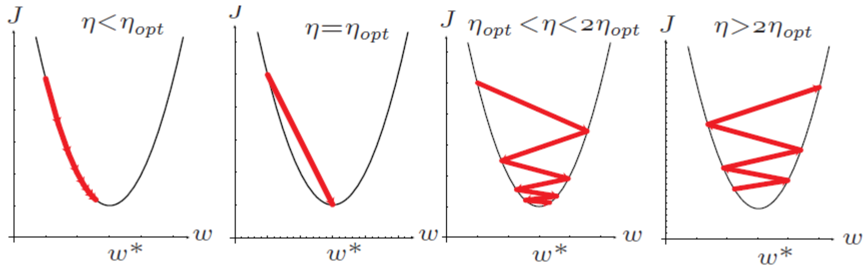
- Polynomial decay
  - $\eta^t \leftarrow \eta^0(1 - t/T)^a$
- Exponential decay
  - $\eta^t \leftarrow \eta^0 a^{-bt}$
- Step decay
  - $\eta^t \leftarrow \eta^0\left(1 - \left\lfloor\frac{t}{b}\right\rfloor/T\right)^a$
  - $\eta^t \leftarrow \eta^0 a^{-\left\lfloor\frac{t}{b}\right\rfloor}$



Adaptive Learning Rates

- Adaptive learning rate dependent on ΔE
  - $\Delta\eta^t \leftarrow \begin{cases} +a & \text{if } E^t < E^{t-1} \\ -b\eta^{t-1} & \text{otherwise} \end{cases}$

    $\eta^t \leftarrow \eta^{t-1} + \Delta\eta^t$
  - $w^{t+1} \leftarrow w^t - \eta^t \dfrac{\partial E^t}{\partial w}$



- Adaptive learning rate dependent on past Δw
  - $w^{t+1} \leftarrow w^t - \dfrac{\eta}{\sqrt{v^t}} \dfrac{\partial E^t}{\partial w}$
  - AdaGrad [Duchi 2011]
    - $v^t \leftarrow v^{t-1} + \left(\dfrac{\partial E^t}{\partial w}\right)^2$; $v^0 = 0$
  - RMSProp [Hinton 2012]
    - $v^t \leftarrow \beta v^{t-1} + (1-\beta)\left(\dfrac{\partial E^t}{\partial w}\right)^2$; $v^0 = 0, \beta = 0.999$

Adaptive Moment Estimation

- Adam [Kingma 2015]
  - $w^{t+1} \leftarrow w^t + \frac{\eta}{\sqrt{\tilde{v}^t + \epsilon}} \Delta\tilde{w}^t$

    $\Delta w^t \leftarrow \alpha \Delta w^{t-1} - (1-\alpha)\frac{\partial E^t}{\partial w}$; $\Delta w^0 = 0, \alpha = 0.9$

    $\Delta\tilde{w}^t \leftarrow \frac{\Delta w^t}{1-(\alpha)^t}$

    $v^t \leftarrow \beta v^{t-1} + (1-\beta)\left(\frac{\partial E^t}{\partial w}\right)^2$; $v^0 = 0, \beta = 0.999$

    $\tilde{v}^t \leftarrow \frac{v^t}{1-(\beta)^t}$

$\Delta w^t \leftarrow \alpha \Delta w^{t-1} - \eta \frac{\partial E^t}{\partial w}$

$\Delta w^t \leftarrow \alpha \Delta w^{t-1} - (1-\alpha)\acute{\eta} \frac{\partial E^t}{\partial w}$

$w^{t+1} \leftarrow w^t + \Delta w^t$

$w^{t+1} \leftarrow w^t - \frac{\eta}{\sqrt{v^t}} \frac{\partial E^t}{\partial w}$

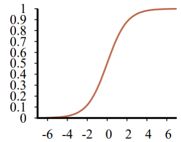
Contents

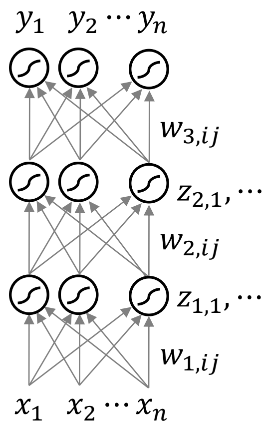
- ❑ Weight Update Schedules
- ❑ Momentum
- ❑ Learning Rates
- ❑ Normalization
- ❑ Regularization
- ❑ Hyperparameters

Batch Normalization

- Batch normalization [Ioffe 2015]
  - $\hat{z}_{l,i} = \alpha_{l,i} \frac{z_{l,i} - \mu_{l,i}}{\sqrt{\sigma_{l,i}^2 + \epsilon}} + \beta_{l,i}$; $s_{l,i} = \boldsymbol{w}_{l,i}^{\top} \boldsymbol{z}_{l-1}$
  - $\mu_{l,i} = \frac{1}{|m|} \sum_{z_{l,i} \in m} z_{l,i}$; m: mini-batch

    $\sigma_{l,i}^2 = \frac{1}{|m|} \sum_{z_{l,i} \in m} \left(z_{l,i} - \mu_{l,i}\right)^2$





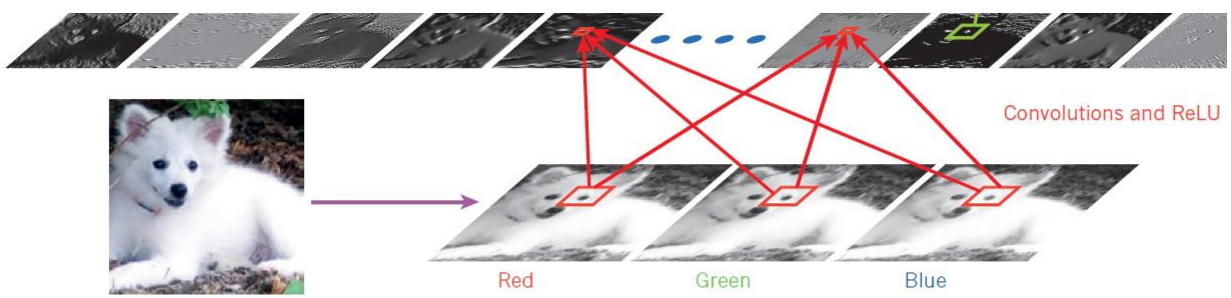
Layer/Instance Normalizations

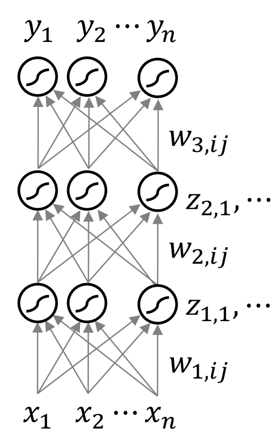
- Layer normalization [Ba 2016]
  - $\hat{s}_{l,i} = \alpha_{l,i} \frac{s_{l,i} - \mu_l}{\sqrt{\sigma_l^2}} + \beta_{l,i}$; $s_{l,i} = \boldsymbol{w}_{l,i}^{\top} \boldsymbol{z}_{l-1}$
  - $\mu_l = \frac{1}{|S_l|} \sum_{s \in S_l} s$; $S_l \equiv \{s_{l,i}\}$

    $\sigma_l^2 = \frac{1}{|S_l|} \sum_{s \in S_l} (s - \mu_l)^2$

- Instance normalization [Ulyanov 2017]
  - $\hat{z}_{l,c,i} = \frac{z_{l,c,i} - \mu_{l,c}}{\sqrt{\sigma_{l,c}^2 + \epsilon}}$; c: channel
  - $\mu_{l,c} = \frac{1}{|\mathcal{Z}_{l,c}|} \sum_{z \in \mathcal{Z}_{l,c}} z$; $\mathcal{Z}_{l,c}$: nodes in layer l and channel c

    $\sigma_{l,c}^2 = \frac{1}{|\mathcal{Z}_{l,c}|} \sum_{z \in \mathcal{Z}_{l,c}} \left(z - \mu_{l,c}\right)^2$





Contents

- ❑ Weight Update Schedules
- ❑ Momentum
- ❑ Learning Rates
- ❑ Normalization
- ❑ Regularization
- ❑ Hyperparameters

Regularization

- Regularization
  - $\textsc{Cost}(h) \equiv \textsc{EmpLoss}_{L,E}(h) + \lambda \overbrace{\textsc{Complexity}(h)}^{\text{regularization function}}$; λ: hyperparameter
  - $\hat{h}^* = \arg\min_{h \in \mathcal{H}} \textsc{Cost}(h)$; $\arg\max_{h \in \mathcal{H}} P(h|data)$

$$= \arg\max_{h \in \mathcal{H}} \underbrace{P(data|h)P(h)}_{\log P(data|h) + \log P(h)}$$

Model Selection Methods

- Regularization
  - $E \equiv$ error of data using the model $+ \lambda \cdot$ model complexity
  - e.g., $E = \sum_t \left(r^t - g(x^t|\boldsymbol{w})\right)^2 + \lambda \frac{1}{2} \sum_i w_i^2$; $g(x|\boldsymbol{w}) = w_n x^n + \cdots + w_1 x + w_0$

    $$\arg\min_{\boldsymbol{w}} \left[\sum_t \left(r^t - g(x^t|\boldsymbol{w})\right)^2 + \lambda \frac{1}{2} \sum_i w_i^2\right]$$

- Minimum description length (MDL)
  - $E \equiv$ description length of data using the model $+$ description length of the model

- Bayesian model selection
  - $P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model}) P(\text{model})}{P(\text{data})}$
  - $\log P(\text{data}|\text{model}) + \log P(\text{model})$
  - $\arg\max_{g} [\log P(\mathcal{D}|g) + \log P(g)]$; $g(x|\boldsymbol{w}) = w_n x^n + \cdots + w_1 x + w_0$

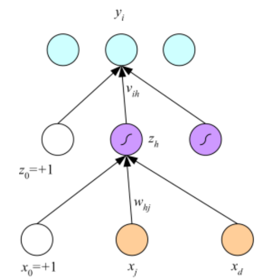
    $= \arg\max_{\boldsymbol{w}} [\log P(\mathcal{D}|\boldsymbol{w}) + \log P(\boldsymbol{w})]$; $P(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{0}, 1/\lambda\, \mathbf{I})$

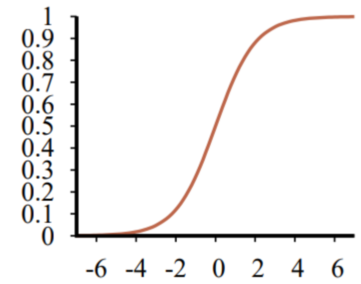
    $= \arg\min_{\boldsymbol{w}} \left[\sum_t \left(r^t - g(x^t|\boldsymbol{w})\right)^2 + \lambda \frac{1}{2} \sum_i w_i^2\right]$ $= \prod_i \frac{1}{\sqrt{2\pi/\lambda}} \exp\left(-\frac{1}{2} \frac{w_i^2}{1/\lambda}\right)$

Weight Decay

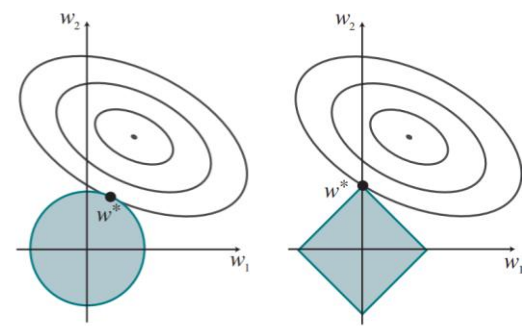
- Weight decay
  - L2 regularization
    - $\acute{E} = E + \frac{\lambda}{2}\sum_i w_i^2$
    - $\Delta w_i \leftarrow -\eta \frac{\partial \acute{E}}{\partial w_i}$
    - $w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i} - \acute{\lambda} w_i$
  - L1 regularization
    - $\acute{E} = E + \lambda \sum_i |w_i|$
    - $\Delta w_i \leftarrow -\eta \frac{\partial \acute{E}}{\partial w_i}$
    - $w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i} - \mathrm{sgn}(w_i)\, \acute{\lambda}$

; $\boldsymbol{z} = \mathbf{W}\boldsymbol{x}$, $\boldsymbol{y} = \mathbf{V}\boldsymbol{z} = \mathbf{V}\mathbf{W}\boldsymbol{x} = \mathbf{U}\boldsymbol{x}$





; $\acute{\lambda} \equiv \eta\lambda$



- Bayesian interpretation of weight decay
  - $\arg\max_{\boldsymbol{w}} P(\boldsymbol{w}|\mathcal{D}) = \arg\max_{\boldsymbol{w}} \log \frac{P(\mathcal{D}|\boldsymbol{w})P(\boldsymbol{w})}{P(\mathcal{D})}$

$$= \arg\max_{\boldsymbol{w}} [\log \underbrace{P(\mathcal{D}|\boldsymbol{w})}_{\prod_t P(\boldsymbol{r}^t|\boldsymbol{x}^t,\boldsymbol{w})P(\boldsymbol{x}^t|\boldsymbol{w})} + \log P(\boldsymbol{w})]$$

$$= \arg\min_{\boldsymbol{w}} \left[E + \frac{\lambda}{2}\sum_i w_i^2\right]$$

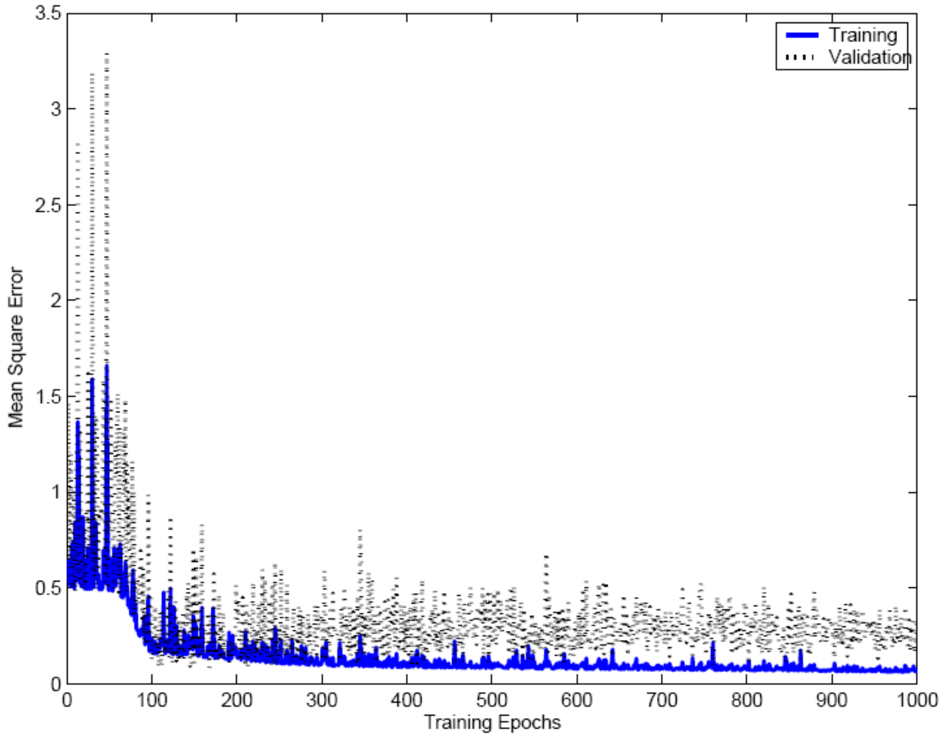
; $-r \log y - (1-r)\log(1-y)$

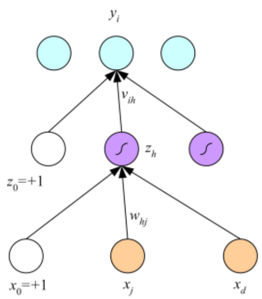
$$P(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \mathbf{0}, 1/\lambda\, \mathbf{I})$$

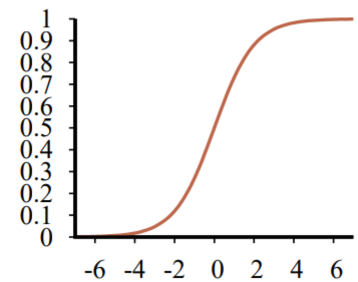
$$= \prod_i \frac{1}{\sqrt{2\pi/\lambda}} \exp\left(-\frac{1}{2}\frac{w_i^2}{1/\lambda}\right)$$

Early Stopping

- Early stopping
  - Stop training to avoid overtraining

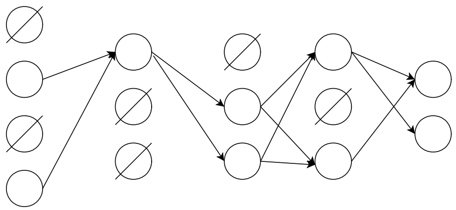






Dropout

- Dropout
  - At each step of training, each unit output is multiplied by a factor of $1/p$ with probability p; otherwise, the unit output is fixed at zero.
  - At inference time, the model is run with no dropout.



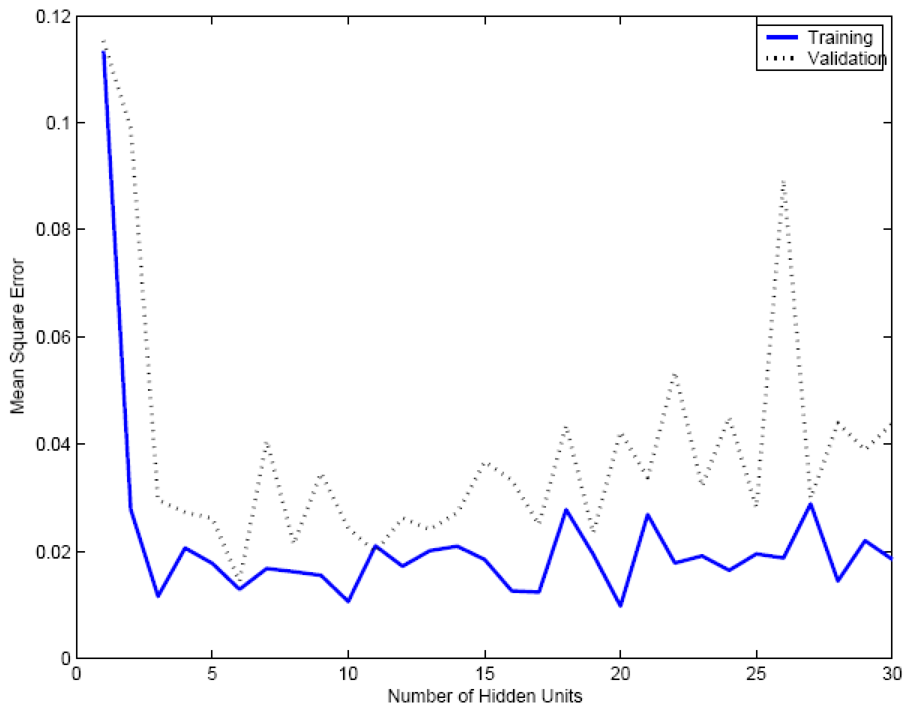
- Why does it work?
  - Noise robust
  - Hidden units compatible with other hidden units
  - Paying attention to all of the abstract features in the later layers
  - Smaller weights
  - A large ensemble of thinned networks

- It is usually necessary to use a larger model and to train it for more iterations.

- Dropconnect

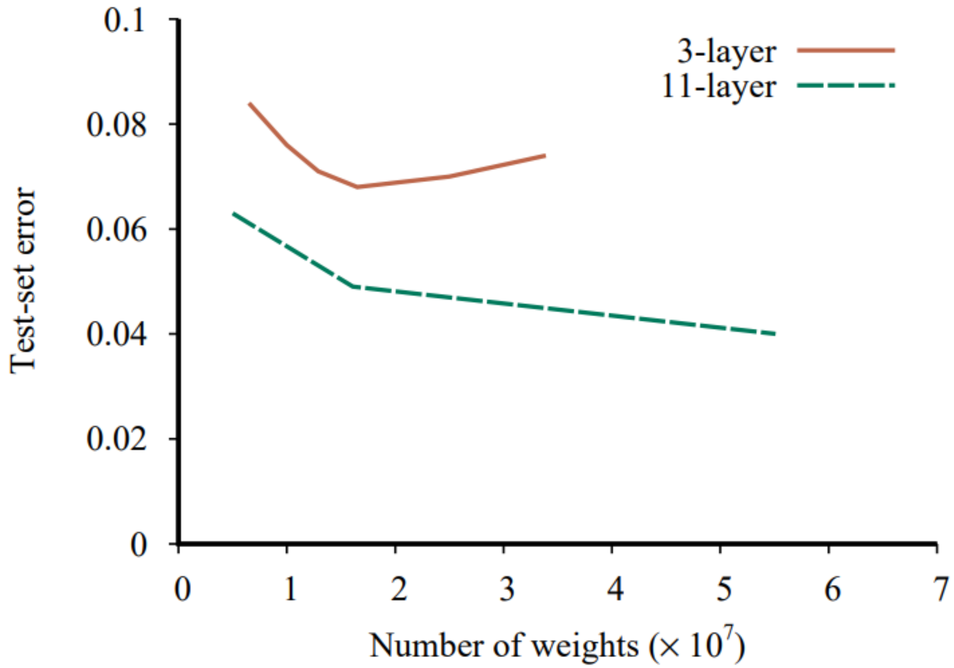
Contents

Number of Hidden Nodes



Network Architectures

- Deeper vs. wider



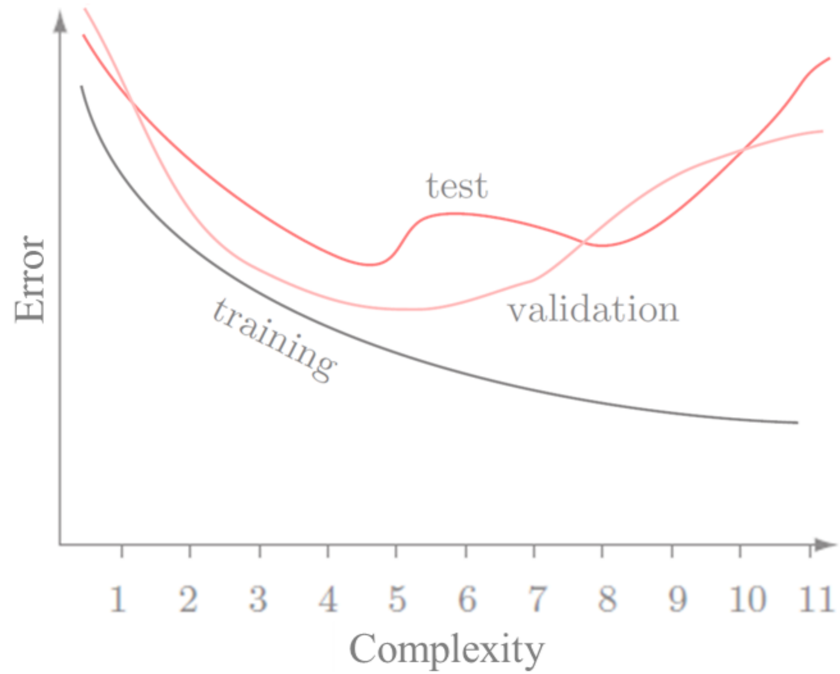
- Network architectures
  - Activation functions
  - Number of nodes in a layer
  - Number of layers
  - Connectivity: e.g., convolution, recurrency, attention, …

Hyperparameters

- Learning rate (η) and learning rate decay schedule
- Momentum: α ; β for Adam
- Mini-batch size and schedule
- Weight decay: λ
- Dropout: p
- Number of epochs ; early stopping
- Architecture
  - Activation functions
  - Number of nodes in a layer
  - Number of layers
  - Connectivity: e.g., convolution, recurrency, attention, …

Cross Validation

- Cross validation



[Duda *et al*. 2001]

- Data sets
  - Training data: for optimizing model *parameters*
  - Validation data (development data): for optimizing *hyperparameters*
  - Test data (publication data, evaluation data): for reporting the final error rate

Hyperparameter Tuning

- Hand-tuning
- Grid search
- Random search
- Bayesian optimization
- Population-based training (PBT)
- Automated machine learning (AutoML)

Summary and Preview

- Weight Update Schedules
  - Batch/Stochastic/Mini-Batch Gradient Descent
- Momentum
  - Momentum
- Learning Rates
  - Polynomial/Exponential/Step Decay, Adaptive, AdaGrad, RMSProp, Adam
- Normalization
  - Batch/Layer/Instance Normalizations
- Regularization
  - Weight Decay, Early Stopping, Dropout
- Hyperparameters
  - η, α, β, $|m|$, λ, p, Epochs, Architecture, …

- Deep Generative Models

References

- Richard Duda, Peter Hart, and David Stork, *Pattern Classification* (2nd edition), Wiley-Interscience, 2001.
- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- Ethem Alpaydin, *Introduction to Machine Learning* (4th edition), Chapters 11 and 12, MIT Press, 2020.
- Stuart Russell and Peter Norvig, *Artificial Intelligence – A Modern Approach* (4th edition), Chapters 19, 21, and 24, Pearson, 2021.
- Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola, *Dive into Deep Learning*, 2019.

- Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *International Conference on Machine Learning*, pp. 448-456, 2015.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, “Layer normalization,” *arXiv*, 2016.
- Dmitry Ulyanov and Andrea Vedaldi, “Instance normalization: The missing ingredient for fast stylization,” *arXiv*, 2017.