Introduction to Deep Learning

5- Training (Part 3) : Performance and Regularization

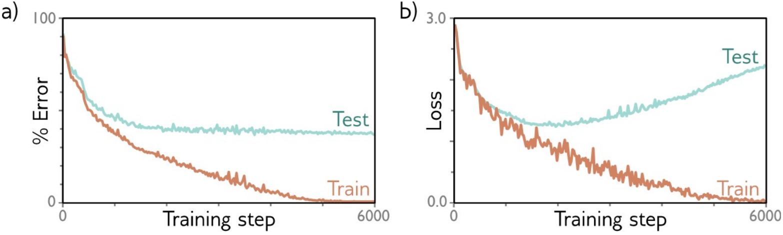
Prof. Monir EL ANNAS

Measuring performance

Sources of error: Example model and performance on MNIST1D dataset

- **Network Configuration:**
  - 40 inputs
  - 10 outputs
  - 4000 training examples (∼400 per class)
  - Two hidden layers, each with 100 units
  - SGD with batch size 100, learning rate 0.1
  - 6000 steps (Epochs)

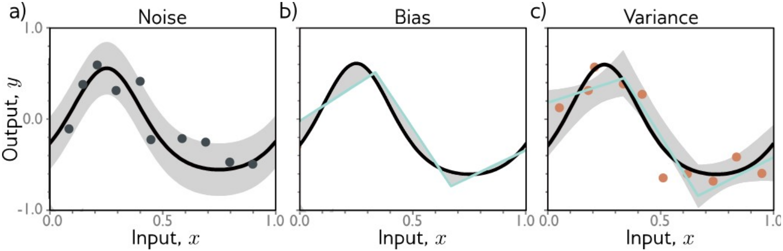
MNIST Dataset



Measuring performance

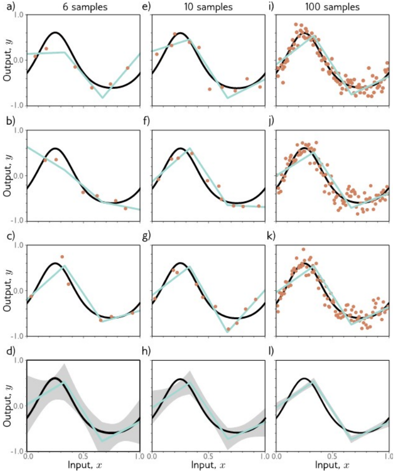
Sources of error: Noise, Bias and Variance

- **Noise** is inherent uncertainty in the true mapping from input to output
- **Bias** is systematic deviation from the mean of the function we are modeling due to limitations in our model
- **Variance** is the uncertainty in fitted model due to choice of training set



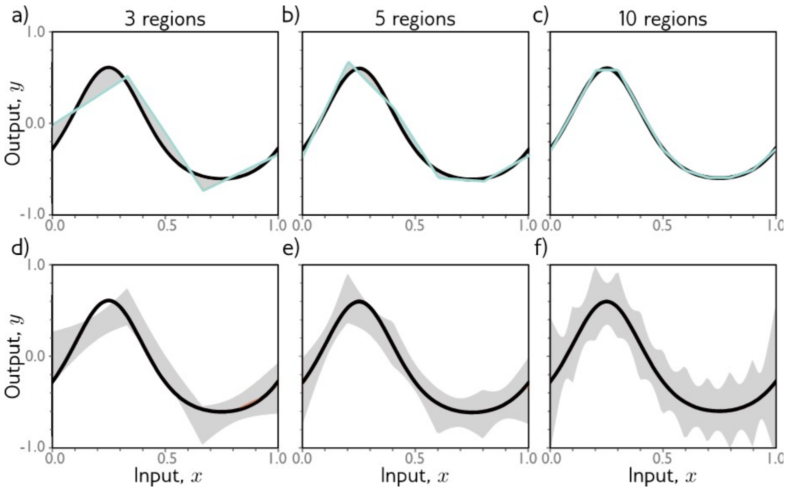
Measuring performance

Reducing the error: Reducing Variance



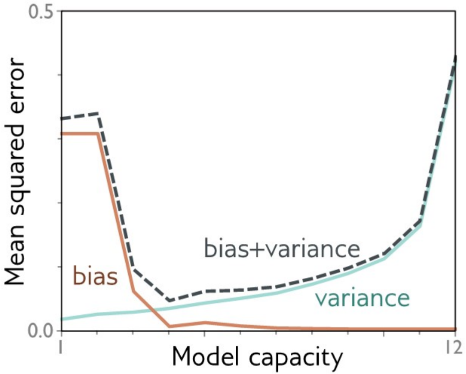
Measuring performance

Reducing the error: Reducing bias



Measuring performance

Reducing the error: Bias-variance trade-off



Measuring performance

Reducing the error: Choosing hyperparameters

- Don't know bias or variance
- Don't know how much capacity to add

- **How do we choose capacity in practice?**
  - Or model structure
  - Or training algorithm
  - Or learning rate
- **Third data set – validation set**
  - Train models with different hyperparameters on training set
  - Choose best hyperparameters with validation set
  - Test once with test set

Regularization

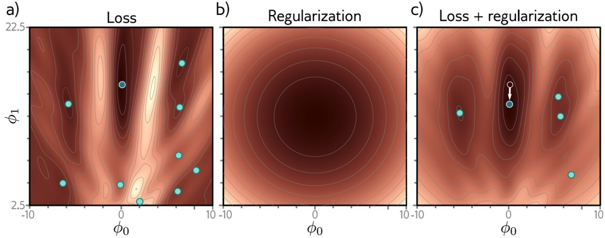
Regularization:

- Why is there a generalization gap between training and test data?
  - Overfitting (model describes statistical peculiarities)
  - Model unconstrained in areas where there are no training examples
- **Regularization** = methods to reduce the generalization gap
- Technically means adding terms to loss function
- But colloquially means any method (hack) to reduce gap

Regularization

Explicit regularization:

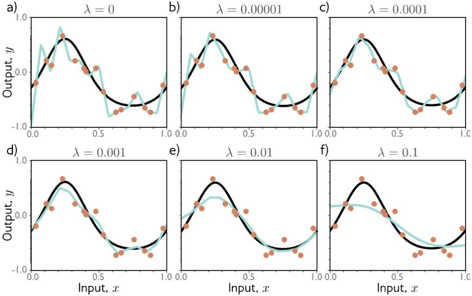
- Standard loss function: $\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[L(\phi)\right] = \underset{\phi}{\operatorname{argmin}} \left[\sum_{i=1}^{I} \ell(x_i, y_i)\right]$
- Regularization adds an extra term $\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[\sum_{i=1}^{I} \ell(x_i, y_i) + \lambda \cdot g(\phi)\right]$
- Favors some parameters, disfavors others.
- $\lambda \geq 0$ controls the strength



Regularization

Explicit regularization: L2 Regularization

- Can only use very general terms
- Most common is L2 regularization
- Favors smaller parameters $\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[L(\phi, \{x_i, y_i\}) + \lambda \sum_j \phi_j^2 \right]$
- Also called Tikhonov regularization, ridge regression
- In neural networks, usually just for weights and called weight decay



Regularization

Implicit regularization

- Gradient descent disfavors areas where gradients are steep

$$\tilde{L}_{GD}[\phi] = L[\phi] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \phi}\right\|^2$$

- SGD likes all batches to have similar gradients

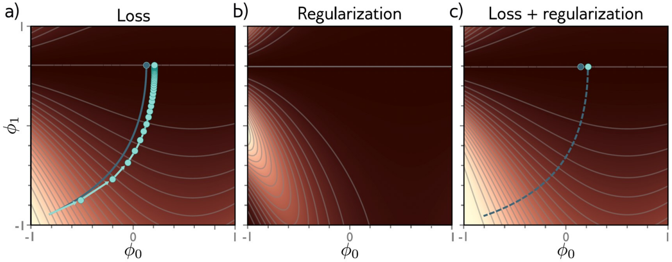
$$\tilde{L}_{SGD}[\phi] = \tilde{L}_{GD}[\phi] + \frac{\alpha}{4B}\sum_{b=1}^{B}\left\|\frac{\partial L_b}{\partial \phi} - \frac{\partial L}{\partial \phi}\right\|^2$$

$$= L[\phi] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \phi}\right\|^2 + \frac{\alpha}{4B}\sum_{b=1}^{B}\left\|\frac{\partial L_b}{\partial \phi} - \frac{\partial L}{\partial \phi}\right\|^2$$

- Depends on learning rate – perhaps why larger learning rates generalize better.

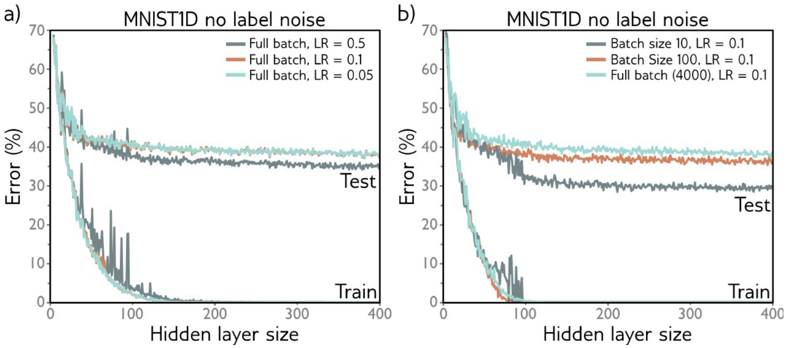
Regularization

Implicit regularization



Regularization

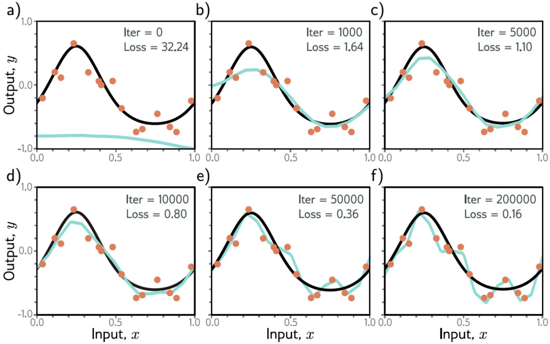
Implicit regularization: MNIST-1D example



Regularization

Early stopping

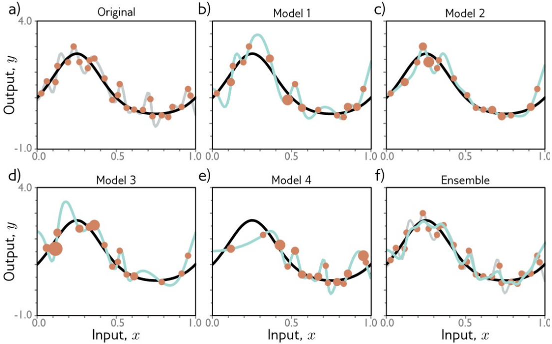
- If we stop training early, weights don't have time to overfit to noise
- Weights start small, don't have time to get large
- Reduces effective model complexity
- Known as early stopping
- Don't have to re-train



Regularization

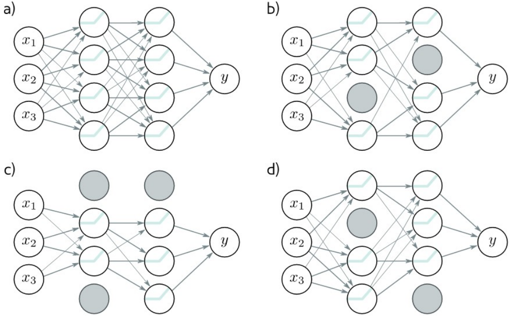
Ensembling

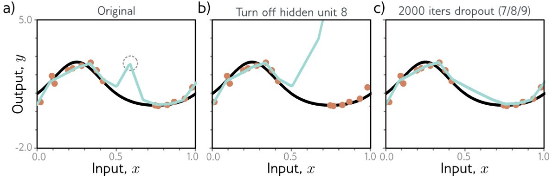
- Average together several models – an ensemble
- Can take mean or median
- Different initializations / different models
- Different subsets of the data resampled with replacements – bagging



Regularization

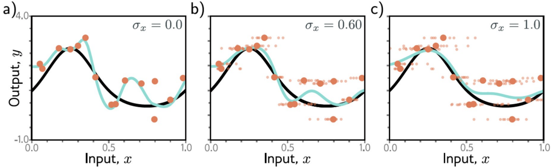
Dropout





Regularization

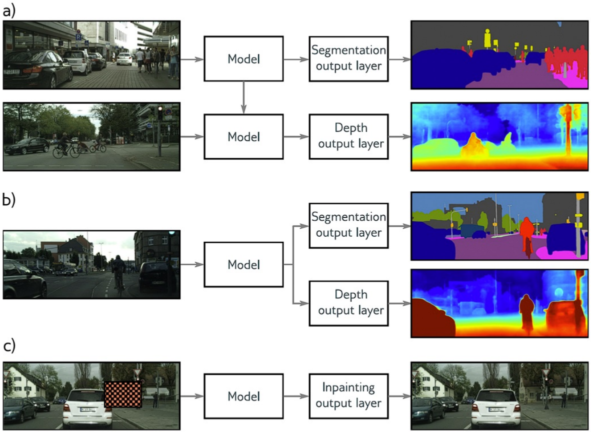
Adding noise



Regularization

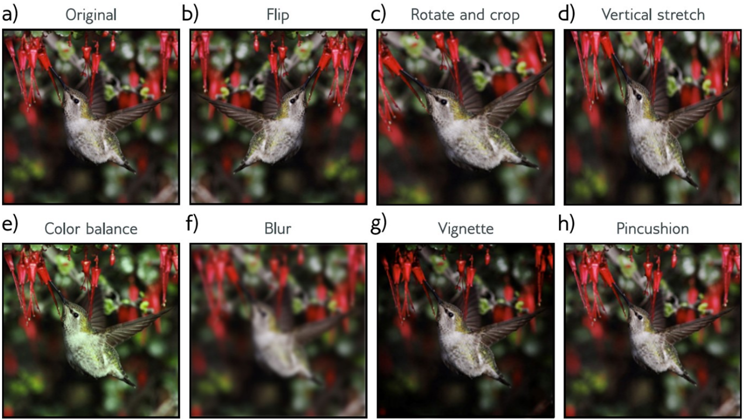
Transfer learning, multi-task learning, self-supervised learning

- Transfer learning
- Multi-task learning
- Self-supervised learning



Regularization

Data augmentation



Regularization

Regularization overview