COS324:
Introduction To Machine Learning

Prof. Yoram Singer
Princeton University

Topic: Generalization and Regularization I

Thus Far

Definitions of learning problems

Linear and non-linear models

Using differentiable loss for learning

Learning algorithms

Thus Far

Definitions of learning problems

Linear and non-linear models

Using differentiable loss for learning

Learning algorithms

Mentioned in passing through examples **test** loss & error

Thus Far

Definitions of learning problems

Linear and non-linear models

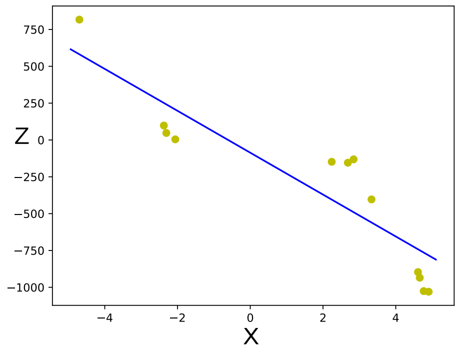
Using differentiable loss for learning

Learning algorithms

Mentioned in passing through examples **test** loss & error

Should the loss/error on unseen data resemble training loss/error ?

Dataset of examples each has two features $\left\{(x_i, z_i)\right\}_{i=1}^{20}$



Learn a function $f : \mathbf{R} \to \mathbf{R}$

Regression loss: $\left(f(x) - z\right)^2$

Choose an order **p** for a polynomial:

$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_p x^p$

Learn coefficients $a_0, a_1, a_2, \ldots, a_p$

Learning Polynomials

Replace $x \mapsto \mathbf{x} = (1, x, x^2, x^3, \ldots, x^p)$

For example suppose $x_i = 3$ and $p = 5$ then $x_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

Learning Polynomials

$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdots \\ \mathbf{x}_n \end{bmatrix}$

Replace $x \mapsto \mathbf{x} = (1, x, x^2, x^3, \ldots, x^p)$

For example suppose $x_i = 3$ and $p = 5$ then $x_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

Learning Polynomials

$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdots \\ \mathbf{x}_n \end{bmatrix}$

Replace $x \mapsto \mathbf{x} = (1, x, x^2, x^3, \ldots, x^p)$

For example suppose $x_i = 3$ and $p = 5$ then $x_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

1	x_1	$(x_1)^2$	…	…	$(x_1)^5$
1	x_2	$(x_2)^2$			…
1	x_3	$(x_3)^2$			…
1	x_4	$(x_4)^2$	…	…	$(x_4)^5$

Learning Polynomials

$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdots \\ \mathbf{x}_n \end{bmatrix}$

Replace $\mathrm{x} \mapsto \mathbf{x} = (1, \mathrm{x}, \mathrm{x}^2, \mathrm{x}^3, \ldots, \mathrm{x}^p)$

For example suppose $\mathrm{x}_i = 3$ and $p = 5$ then $\mathrm{x}_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

1	x_1	$(x_1)^2$	…	…	$(x_1)^5$
1	x_2	$(x_2)^2$			…
1	x_3	$(x_3)^2$			…
1	x_4	$(x_4)^2$	…	…	$(x_4)^5$

a_1
a_2
a_3
a_4
a_5

Learning Polynomials

$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdots \\ \mathbf{x}_n \end{bmatrix}$

Replace $\mathrm{x} \mapsto \mathbf{x} = (1, \mathrm{x}, \mathrm{x}^2, \mathrm{x}^3, \ldots, \mathrm{x}^p)$

For example suppose $\mathrm{x}_i = 3$ and $p = 5$ then $\mathrm{x}_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

1	x_1	$(x_1)^2$	…	…	$(x_1)^5$
1	x_2	$(x_2)^2$			…
1	x_3	$(x_3)^2$			…
1	x_4	$(x_4)^2$	…	…	$(x_4)^5$

a_1
a_2
a_3
a_4
a_5

$\approx$

z_1
z_2
z_3
z_4

Learning Polynomials

$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdots \\ \mathbf{x}_n \end{bmatrix}$

Replace $\mathrm{x} \mapsto \mathbf{x} = (1, \mathrm{x}, \mathrm{x}^2, \mathrm{x}^3, \ldots, \mathrm{x}^p)$

For example suppose $\mathrm{x}_i = 3$ and $p = 5$ then $\mathrm{x}_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

1	x_1	$(x_1)^2$	…	…	$(x_1)^5$
1	x_2	$(x_2)^2$			…
1	x_3	$(x_3)^2$			…
1	x_4	$(x_4)^2$	…	…	$(x_4)^5$

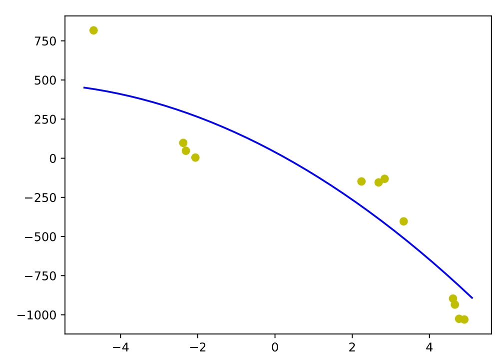
a_1
a_2
a_3
a_4
a_5

$\approx$

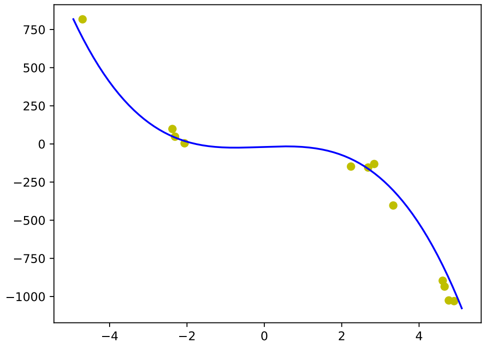
z_1
z_2
z_3
z_4

$$\min_{\mathbf{a}} \|X\mathbf{a} - \mathbf{z}\|^2$$

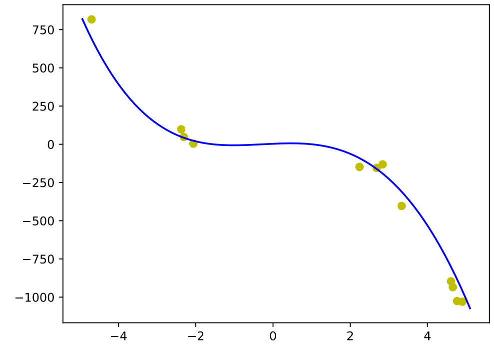
Degree 2 Fit to Training Data



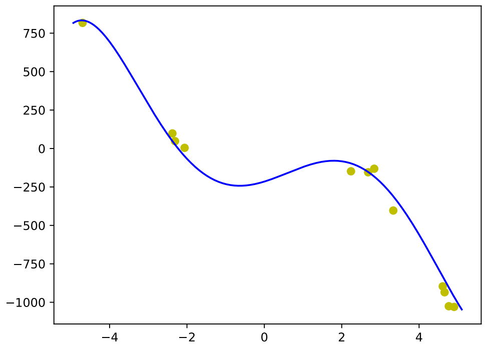
Degree 3 Fit to Training Data



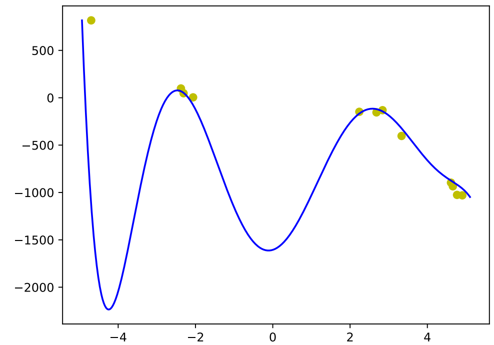
Degree 4 Fit to Training Data



Degree 5 Fit to Training Data



Degree 7 Fit to Training Data

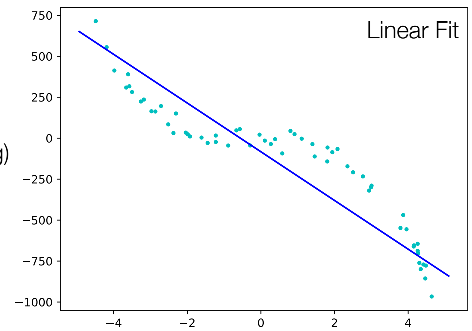


Test Data

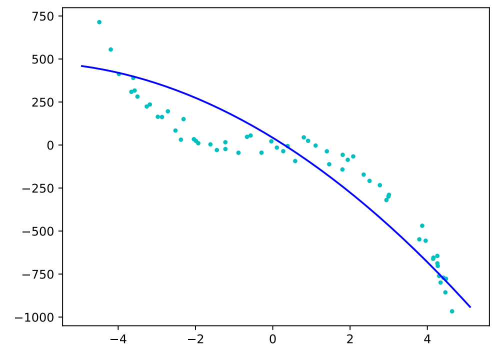
Received many more examples

$$\{(x_i, z_i)\}_{i=1}^{200}$$

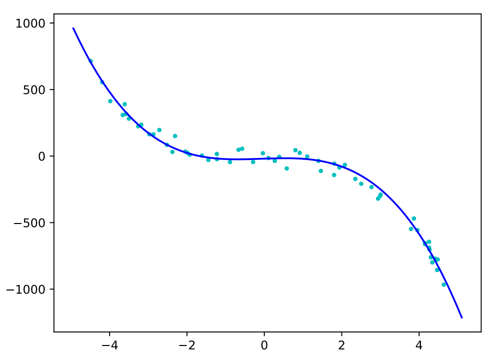
Tested fit on unseen (during training) examples



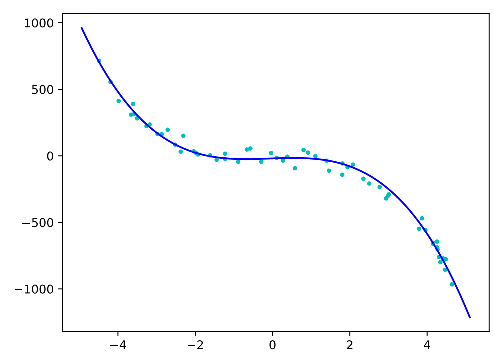
Degree 2 Fit to Test Data



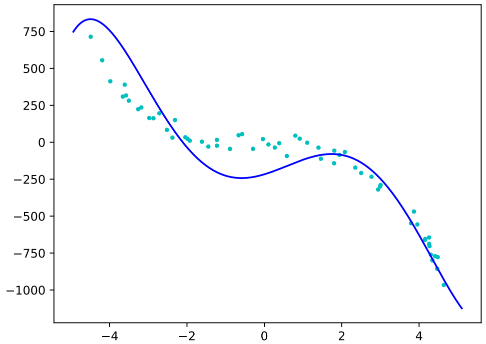
Degree 3 Fit to Test Data



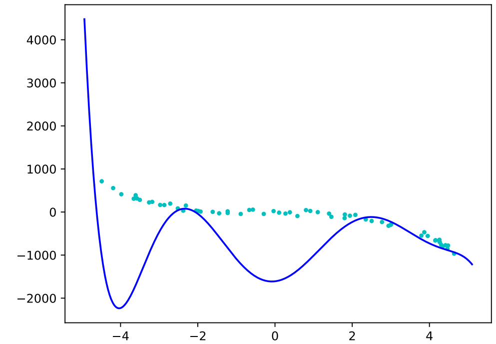
Degree 3 Fit to Test Data

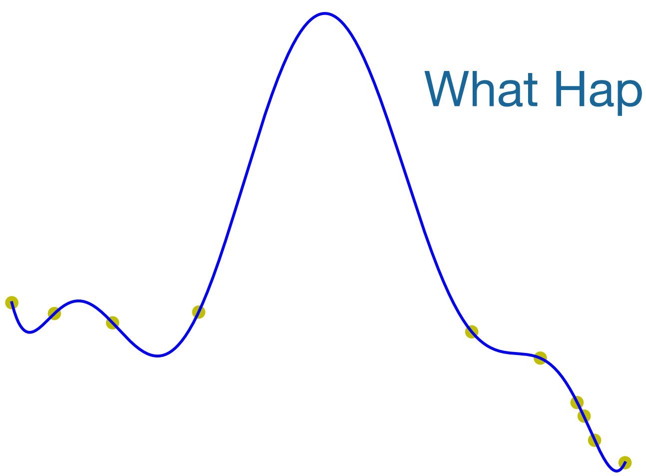


Degree 4 Fit to Test Data

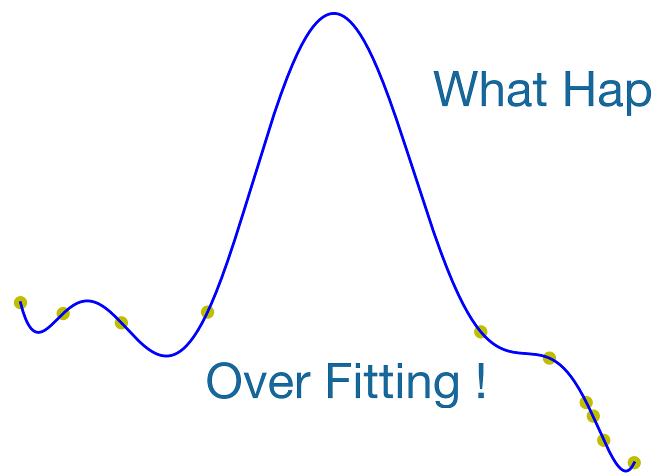


Degree 7 Fit to Test Data





What Happened ?

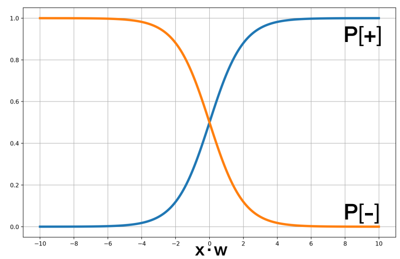


What Happened ?

Over Fitting !

Reminder: Logistic Regression

- Given $\mathbf{x}$ "probability" of y to be +1: $\mathbf{P}\left[+1 \,|\, \mathbf{x}; \mathbf{w}\right] = \dfrac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{x}}}$
- Probability of y to be -1: $\mathbf{P}\left[-1 \,|\, \mathbf{x}; \mathbf{w}\right] = 1 - \dfrac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{x}}} = \dfrac{1}{1 + e^{\mathbf{w}\cdot\mathbf{x}}}$



Overfitting in Logistic Regression

Trained 2 logit models:

$$\mathbf{P}\left[\mathrm{y} \,|\, \mathbf{x}; \mathbf{w}_{\mathrm{j}}\right] = \frac{1}{1 + e^{-\mathrm{y}\,\mathbf{w}_{\mathrm{j}}\cdot\mathbf{x}}} \quad \mathrm{j} \in [2]$$

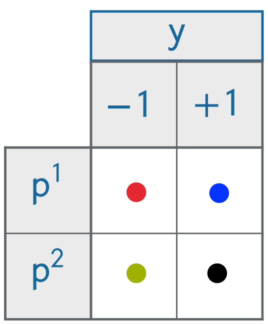
Trained with log-loss: for $(\mathbf{x}_{\mathrm{i}}, \mathrm{y}_{\mathrm{i}})$ loss is $-\log\left(\mathbf{P}\left[\mathrm{y}_{\mathrm{i}} \,|\, \mathbf{x}_{\mathrm{i}}; \mathbf{w}_{\mathrm{j}}\right]\right)$

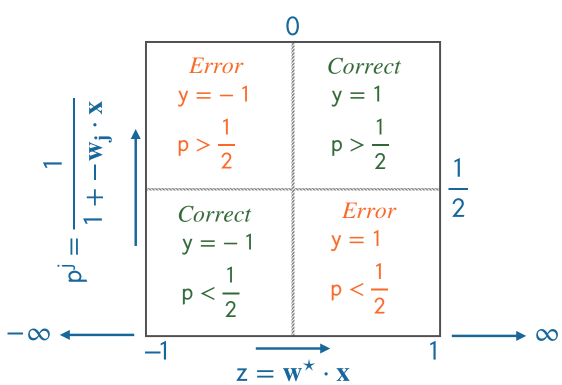
First model was training while guarding for overfitting (more later)

Second model was trained using SGD *without* projections

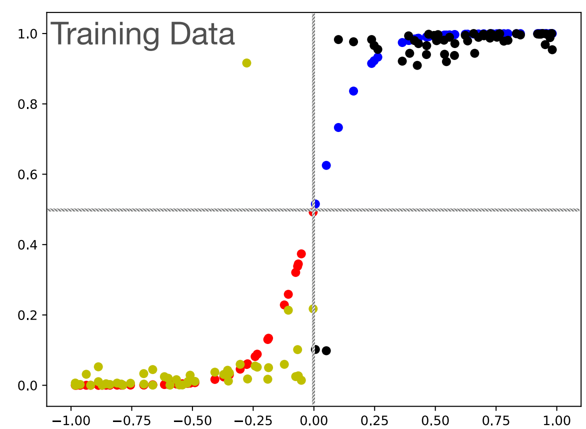
Predictions: **red** & **blue** first model ; **black** & **yellowish** second model

Legend for Graphs

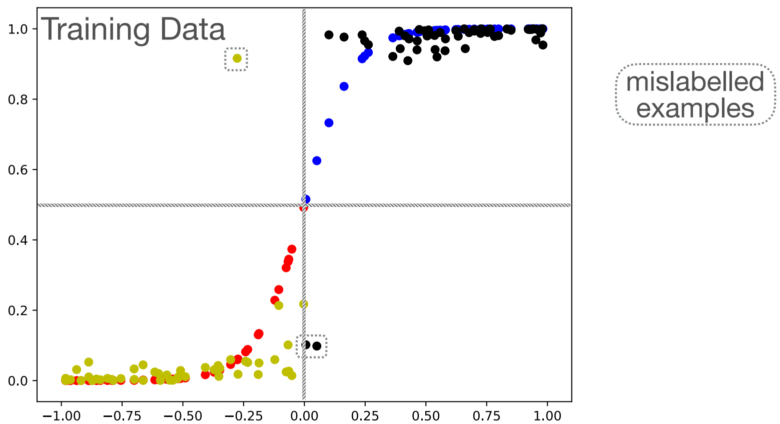




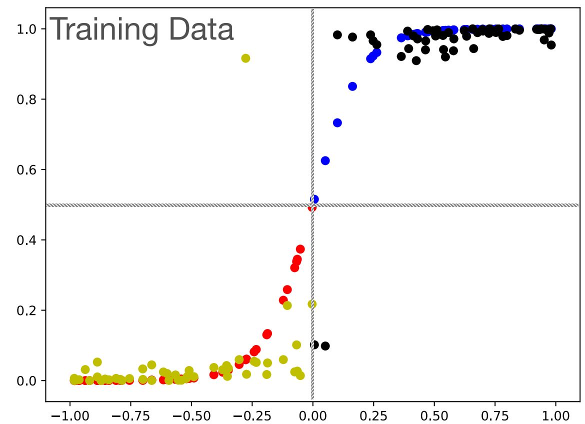
Overfitting in Logistic Regression



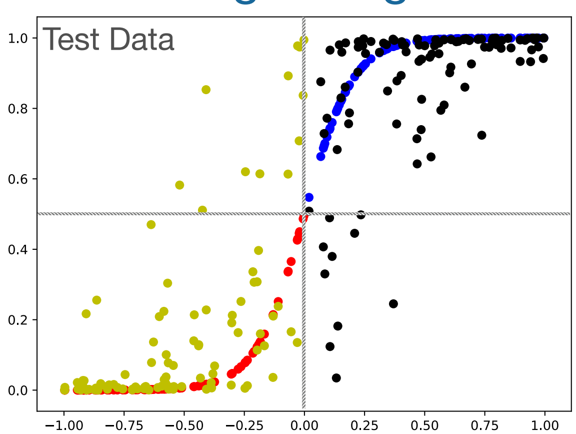
Overfitting in Logistic Regression



Overfitting in Logistic Regression



Overfitting in Logistic Regression



Overfitting in Classification

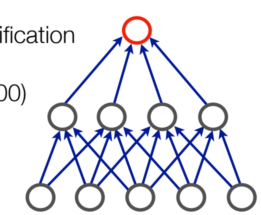
Trained a two-layer NN on binary image classification

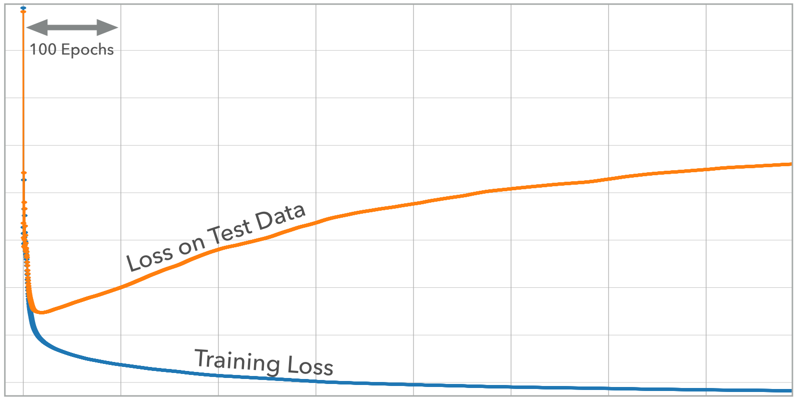
Thumbnail images: 10x10 (input dimension 100)

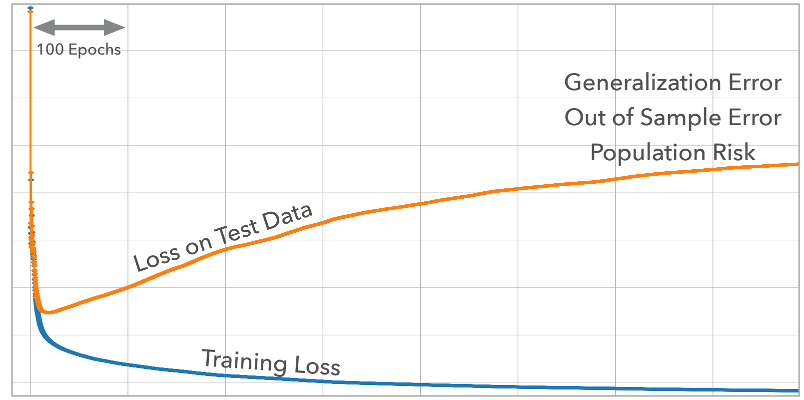
Dataset size 10,000

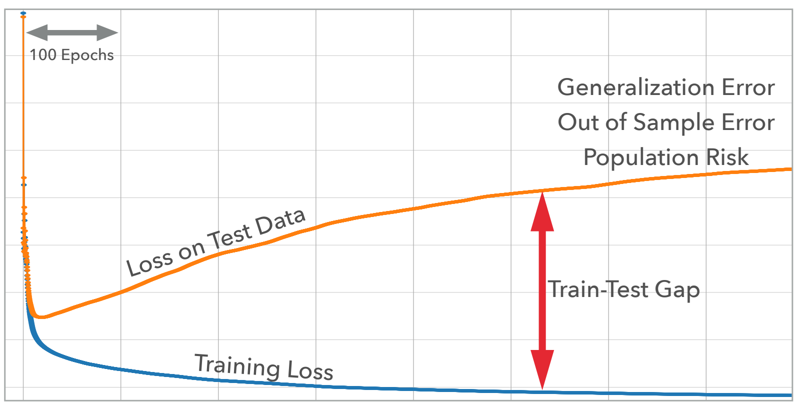
Hidden layer size: 20

Tuned SGD well and ran for many iterations



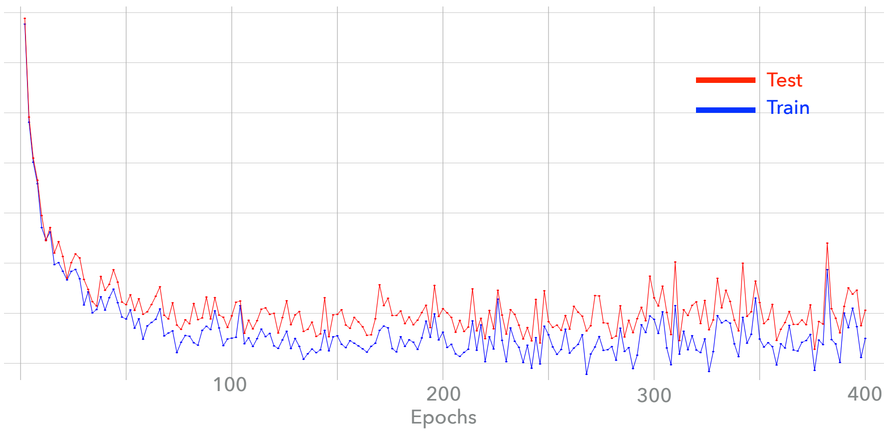




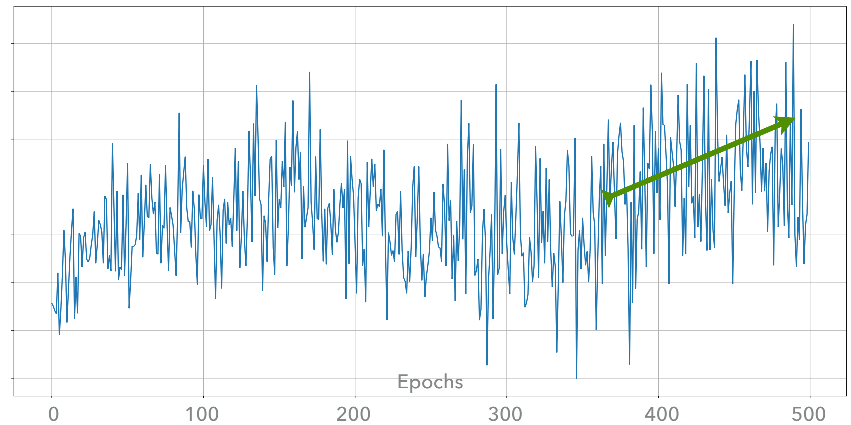


Early Stopping

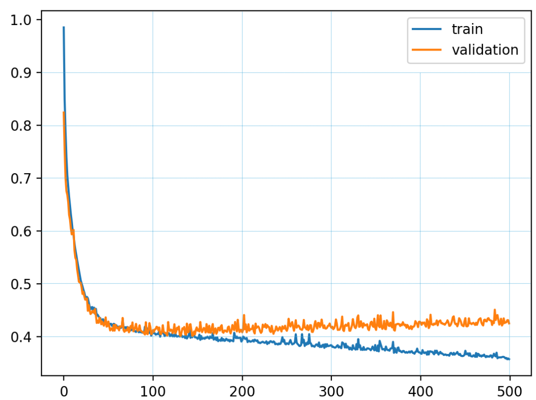
- Use a validation set which is not used for training
- Check every **k** updates/epochs performance on validation set
- Once test-train gap is growing stop training
- Works well in practice when scheme is feasible
  - Requires three sets of examples: Train, Validation, Test
  - Loss of stochastic methods not monotone & gap not easy to monitor

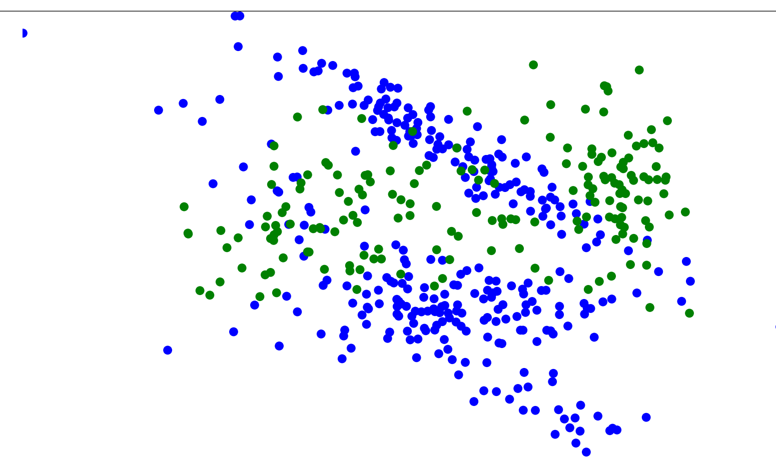


Test-Train Gap

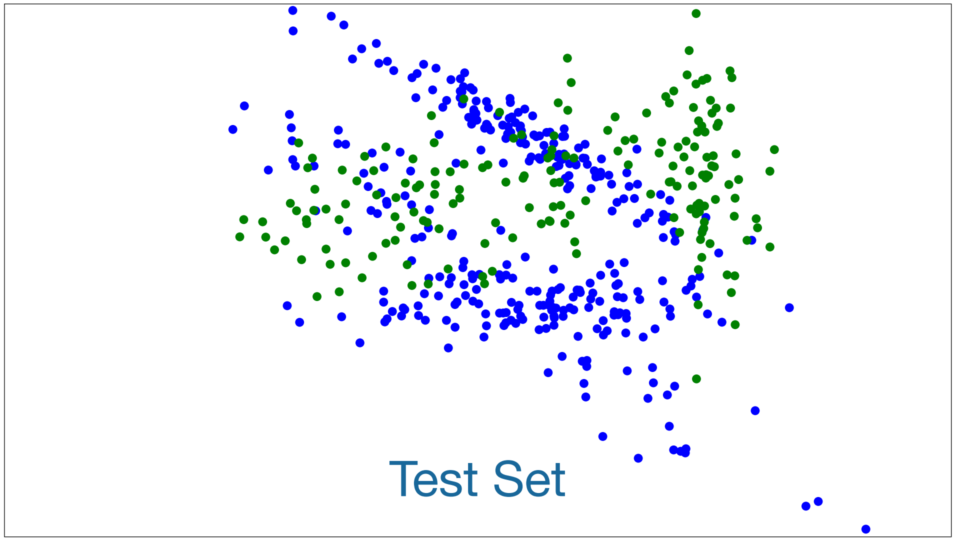


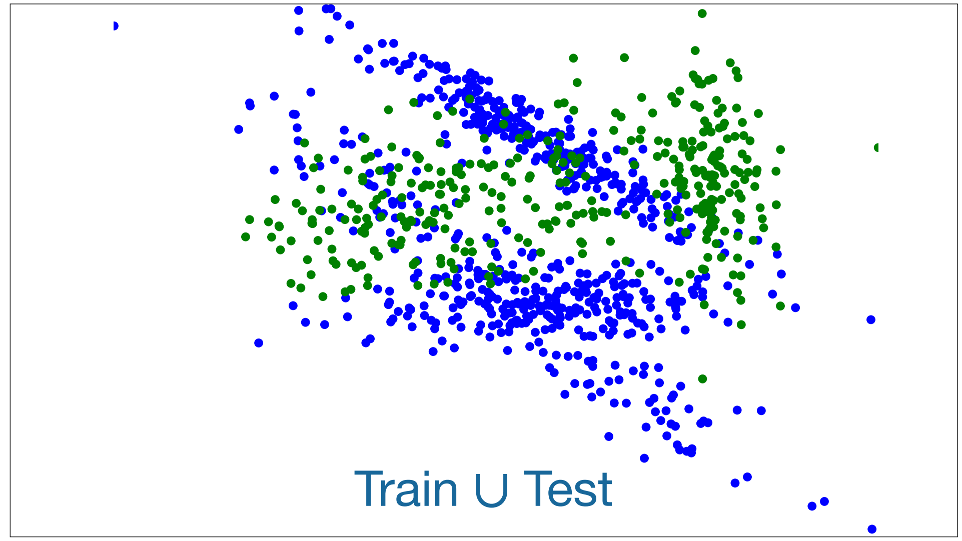
Train-Test Gap (large mini-batch)

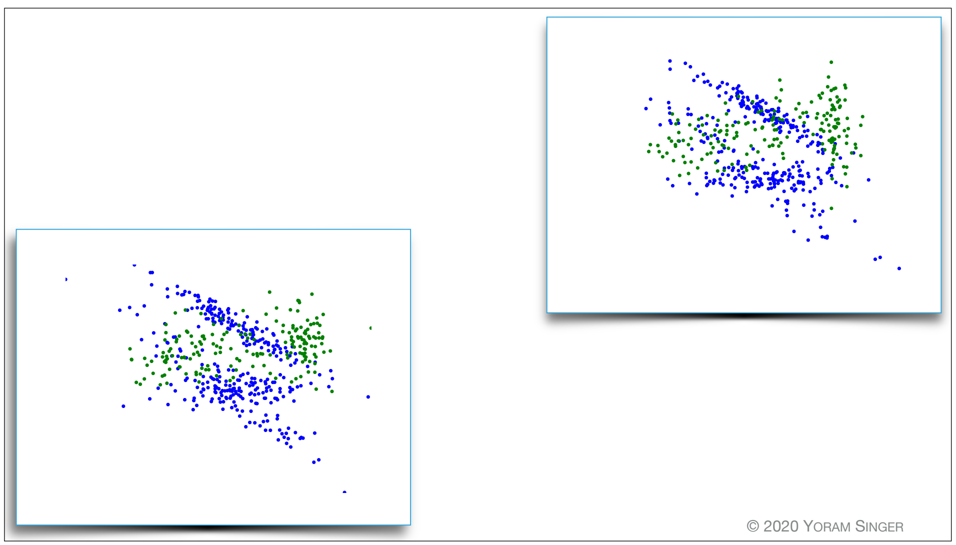


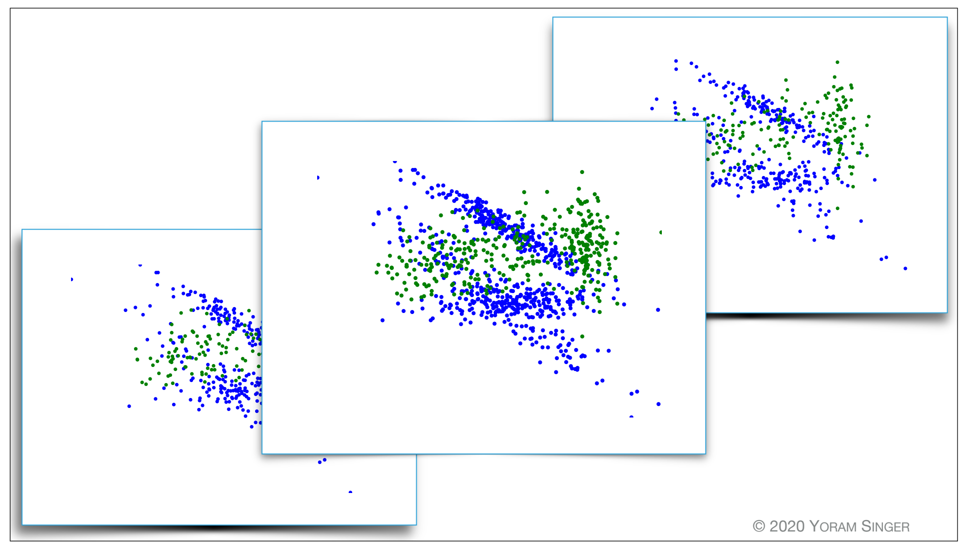


Train Set


Test Set

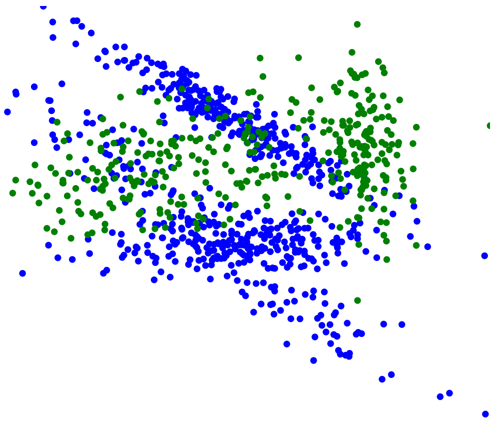

Train ∪ Test


© 2020 Yoram Singer


© 2020 Yoram Singer

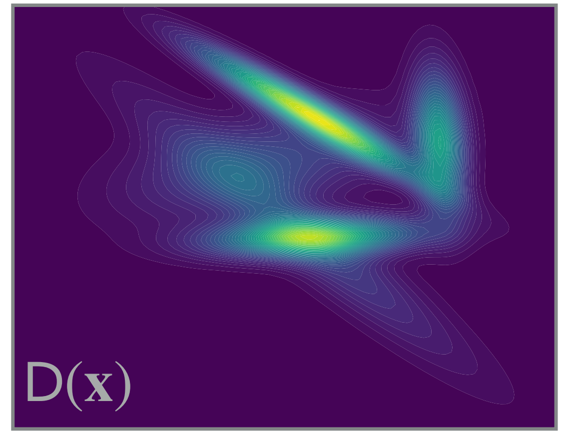
Underlying Distribution

$\mathrm{D}(\mathbf{x}, \mathbf{y}) =$
$\mathrm{D}(\mathbf{x})\,\mathrm{D}(\mathbf{y} \,|\, \mathbf{x})$



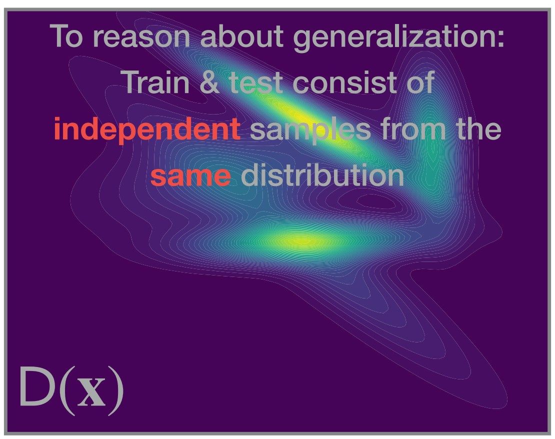
Underlying Distribution

$\mathrm{D}(\mathbf{x}, \mathbf{y}) =$
$\mathrm{D}(\mathbf{x})\,\mathrm{D}(\mathbf{y} \,|\, \mathbf{x})$



Underlying Distribution

$\mathrm{D}(\mathbf{x}, \mathbf{y}) =$
$\mathrm{D}(\mathbf{x})\,\mathrm{D}(\mathbf{y} \,|\, \mathbf{x})$



I.I.D Samples

- I.I.D: Identically Independently Distributed
- Generalization analysis typically assumes $\exists \mathrm{D}$:

  unknown distribution D($\mathbf{x}$,y)
- W.L.O.G assume $\mathbf{x} \in \{0,1\}^d \quad y \in \{-1,1\}$
- Identically [no dependence on i]:

  $\forall i \in S:\;$ $\mathrm{D}((\mathbf{x}_i, y_i) = (\mathbf{a}, b))$ is $\mathrm{D}(\mathbf{a}, b)$
- Independence:

  $$\mathrm{D}\big((\mathbf{x}_i, y_i) = (\mathbf{a}, b) \,\wedge\, \mathrm{D}(\mathbf{x}_j, y_j) = (\mathbf{a}', b')\big) = \mathrm{D}(\mathbf{a}, b)\,\mathrm{D}(\mathbf{a}', b')$$

x_0	x_1	y	D(x,y)
0	0	-1	0.07
0	0	1	0.01
0	1	-1	0.03
…	…	…	…
…	…	…	…
1	1	1	0.005

Generalization Error (deterministic)

Unknown distribution D(**x**)

Generalization Error (deterministic)

Unknown distribution D(**x**)

Deterministic outcome y given **x**: $D(y=-1|\mathbf{x}) = 1$ or $D(y=1|\mathbf{x}) = 1$

$$\Rightarrow h^\star(\mathbf{x}) = 2D(y\,|\,\mathbf{x}) - 1$$

© 2020 YORAM SINGER 34

Generalization Error (deterministic)

Unknown distribution D(**x**)

Deterministic outcome y given **x**: $D(y=-1|\mathbf{x}) = 1$ or $D(y=1|\mathbf{x}) = 1$

$$\Rightarrow h^\star(\mathbf{x}) = 2D(y\,|\,\mathbf{x}) - 1$$

Deterministic predictor $f : \{0,1\}^d \to \{-1,1\}$

© 2020 YORAM SINGER 34

Generalization Error (deterministic)

Unknown distribution D(**x**)

Deterministic outcome y given **x**: $D(y=-1|\mathbf{x}) = 1$ or $D(y=1|\mathbf{x}) = 1$

$$\Rightarrow h^\star(\mathbf{x}) = 2D(y\,|\,\mathbf{x}) - 1$$

Deterministic predictor $f : \{0,1\}^d \to \{-1,1\}$

Generalization error of **f** :

$$\mathrm{err}_D(f) = \sum_{\mathbf{x}} D(\mathbf{x})\, \mathbf{1}[f(\mathbf{x}) \neq h^\star(\mathbf{x})]$$

© 2020 YORAM SINGER 34

Generalization Error (deterministic)

Unknown distribution D(**x**)

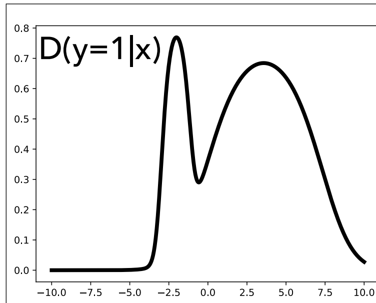
Deterministic outcome y given **x**: $D(y{=}-1|\mathbf{x}) = 1$ or $D(y{=}1|\mathbf{x}) = 1$

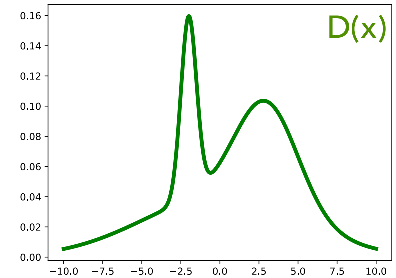
$$\Rightarrow h^\star(\mathbf{x}) = 2D(y\,|\,\mathbf{x}) - 1$$

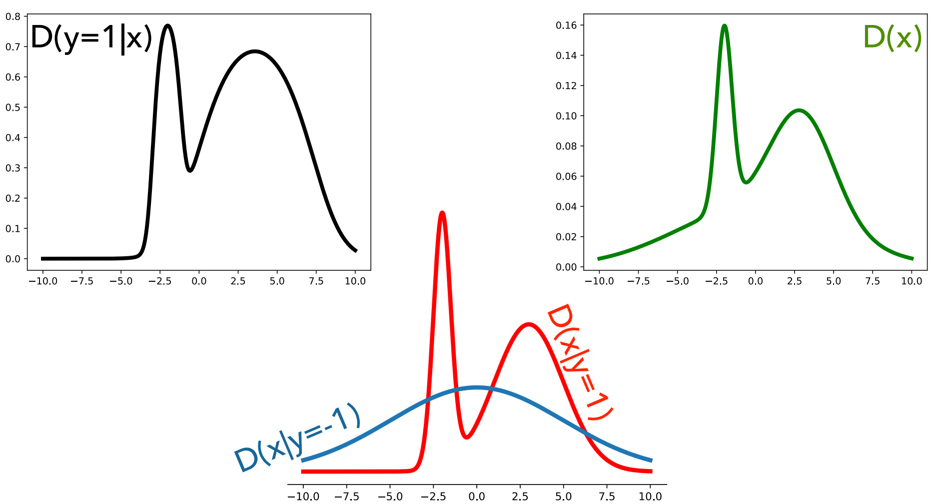
Deterministic predictor $f : \{0,1\}^d \to \{-1,1\}$

Generalization error of **f** :

$$\mathrm{err}_D(f) = \sum_{\mathbf{x}} D(\mathbf{x})\,\mathbf{1}[f(\mathbf{x}) \neq h^\star(\mathbf{x})] = \sum_{\mathbf{x}} D(\mathbf{x}) \sum_{y\in\{-1,1\}} \mathbf{1}[f(\mathbf{x}) \neq y]\, D(y\,|\,\mathbf{x})$$







Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $f : \{0,1\}^d \to \{-1,1\}$

Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $\mathrm{f} : \{0,1\}^d \rightarrow \{-1,1\}$

Given **x** true label *y* is 1 w.p. D(y=1|x) and -1 w.p. D(y=-1|x)

Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $\mathrm{f} : \{0,1\}^d \rightarrow \{-1,1\}$

Given **x** true label *y* is 1 w.p. D(y=1|x) and -1 w.p. D(y=-1|x)

f(**x**)=1 $\Rightarrow$ w.p. D(y=-1|**x**) prediction error ; f(**x**)=-1 $\Rightarrow$ w.p. D(y=1|**x**) prediction error

Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $\mathrm{f} : \{0,1\}^d \rightarrow \{-1,1\}$

Given **x** true label *y* is 1 w.p. D(y=1|x) and -1 w.p. D(y=-1|x)

f(**x**)=1 $\Rightarrow$ w.p. D(y=-1|**x**) prediction error ; f(**x**)=-1 $\Rightarrow$ w.p. D(y=1|**x**) prediction error

Expected error of **f** on **x** : $\mathrm{D}\big(-\mathrm{f}(\mathbf{x})\,|\,\mathbf{x}\big) = \sum_{y\in\{-1,1\}} \mathbf{1}[\mathrm{f}(\mathbf{x}) \neq y]\ \mathrm{D}(y\,|\,\mathbf{x})$

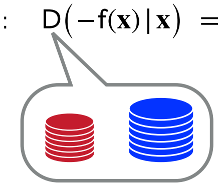
Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $\mathrm{f} : \{0,1\}^d \rightarrow \{-1,1\}$

Given **x** true label *y* is 1 w.p. D(y=1|x) and -1 w.p. D(y=-1|x)

f(**x**)=1 $\Rightarrow$ w.p. D(y=-1|**x**) prediction error ; f(**x**)=-1 $\Rightarrow$ w.p. D(y=1|**x**) prediction error

Expected error of **f** on **x** : $\mathrm{D}\big(-\mathrm{f}(\mathbf{x})\,|\,\mathbf{x}\big) = \sum_{y\in\{-1,1\}} \mathbf{1}[\mathrm{f}(\mathbf{x}) \neq y]\ \mathrm{D}(y\,|\,\mathbf{x})$



Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $f : \{0,1\}^d \rightarrow \{-1,1\}$

Given **x** true label *y* is 1 w.p. D(y=1|x) and -1 w.p. D(y=-1|x)

f(**x**)=1 $\Rightarrow$ w.p. D(y=-1|**x**) prediction error ; f(**x**)=-1 $\Rightarrow$ w.p. D(y=1|**x**) prediction error

Expected error of **f** on **x** : $D\big(-f(\mathbf{x})\,|\,\mathbf{x}\big) \;=\; \sum_{y\in\{-1,1\}} \mathbf{1}[f(\mathbf{x}) \neq y]\; D(y\,|\,\mathbf{x})$

Generalization Error (stochastic)

Unknown distribution D(**x**,y) & deterministic predictor $f : \{0,1\}^d \rightarrow \{-1,1\}$

Given **x** true label *y* is 1 w.p. D(y=1|x) and -1 w.p. D(y=-1|x)

f(**x**)=1 $\Rightarrow$ w.p. D(y=-1|**x**) prediction error ; f(**x**)=-1 $\Rightarrow$ w.p. D(y=1|**x**) prediction error

Expected error of **f** on **x** : $D\big(-f(\mathbf{x})\,|\,\mathbf{x}\big) \;=\; \sum_{y\in\{-1,1\}} \mathbf{1}[f(\mathbf{x}) \neq y]\; D(y\,|\,\mathbf{x})$

Generalization error of **f** :

$$\mathrm{err}_D(f) = \sum_{\mathbf{x}} D(\mathbf{x}) \sum_{y\in\{-1,1\}} \mathbf{1}[f(\mathbf{x}) \neq y]\; D(y\,|\,\mathbf{x})$$

Finite Set of Predictors

Suppose we have only **k** predictors — no weight learning: $f_1, \ldots, f_k$

Finite Set of Predictors

Suppose we have only **k** predictors — no weight learning: $f_1, \ldots, f_k$

One **f** has zero generalization error, rest have generalization error $\geq \epsilon$:

$$\exists j : \forall(\mathbf{x}, y) :\; f_j(\mathbf{x}) = f^\star(\mathbf{x}) = y \quad ; \quad \forall i \neq j : \mathbf{P}[f(\mathbf{x}) \neq y] \geq \epsilon$$

Finite Set of Predictors

Suppose we have only **k** predictors — no weight learning: $f_1, \ldots, f_k$

One **f** has zero generalization error, rest have generalization error $\geq \epsilon$:

$$\exists j : \forall (\mathbf{x}, y) : \; f_j(\mathbf{x}) = f^\star(\mathbf{x}) = y \quad ; \quad \forall i \neq j : \mathbf{P}[f(\mathbf{x}) \neq y] \geq \epsilon$$

Received training set S with only **n** examples sampled independently

Finite Set of Predictors

Suppose we have only **k** predictors — no weight learning: $f_1, \ldots, f_k$

One **f** has zero generalization error, rest have generalization error $\geq \epsilon$:

$$\exists j : \forall (\mathbf{x}, y) : \; f_j(\mathbf{x}) = f^\star(\mathbf{x}) = y \quad ; \quad \forall i \neq j : \mathbf{P}[f(\mathbf{x}) \neq y] \geq \epsilon$$

Received training set S with only **n** examples sampled independently

Evaluate errors on S: $\text{err}_S(f_i) = \epsilon_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[f_i(\mathbf{x}) \neq y_i\right]$

Finite Set of Predictors

Suppose we have only **k** predictors — no weight learning: $f_1, \ldots, f_k$

One **f** has zero generalization error, rest have generalization error $\geq \epsilon$:

$$\exists j : \forall (\mathbf{x}, y) : \; f_j(\mathbf{x}) = f^\star(\mathbf{x}) = y \quad ; \quad \forall i \neq j : \mathbf{P}[f(\mathbf{x}) \neq y] \geq \epsilon$$

Received training set S with only **n** examples sampled independently

Evaluate errors on S: $\text{err}_S(f_i) = \epsilon_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[f_i(\mathbf{x}) \neq y_i\right]$

Choose any f_j for which $\epsilon_j = 0$

Generalization: Finite Case I

Probability that $\epsilon_i = 0$ is at most $(1 - \epsilon)^n \leq e^{-\epsilon n}$ [independence of sample]

Generalization: Finite Case I

Probability that $\epsilon_i = 0$ is at most $(1-\epsilon)^n \leq e^{-\epsilon n}$ [independence of sample]

Probability α that $\exists i \neq j \text{ s.t. } \epsilon_i = 0$ is at most $\alpha = (k-1)\, e^{-\epsilon n}$

Generalization: Finite Case I

Probability that $\epsilon_i = 0$ is at most $(1-\epsilon)^n \leq e^{-\epsilon n}$ [independence of sample]

Probability α that $\exists i \neq j \text{ s.t. } \epsilon_i = 0$ is at most $\alpha = (k-1)\, e^{-\epsilon n}$

If $\alpha \leq \frac{1}{k}$ it is unlikely we do not find correct predictor: $(k-1)\, e^{-\epsilon n} \geq \frac{1}{k}$

Generalization: Finite Case I

Probability that $\epsilon_i = 0$ is at most $(1-\epsilon)^n \leq e^{-\epsilon n}$ [independence of sample]

Probability α that $\exists i \neq j \text{ s.t. } \epsilon_i = 0$ is at most $\alpha = (k-1)\, e^{-\epsilon n}$

If $\alpha \leq \frac{1}{k}$ it is unlikely we do not find correct predictor: $(k-1)\, e^{-\epsilon n} \geq \frac{1}{k}$

This means that we need about $O\left(\frac{\log(k)}{\epsilon}\right)$ samples

Generalization: Finite Case II

Generalization: Finite Case II

Almost always, prefect predictor does not exist: $\epsilon^\star = \min_i \mathrm{err}_D\big(f_i(x)\big) > 0$

Generalization: Finite Case II

Almost always, prefect predictor does not exist: $\epsilon^\star = \min_i \mathrm{err}_D\big(f_i(x)\big) > 0$

Evaluate errors on S: $\mathrm{err}_S(f_i) = \epsilon_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[f_i(\mathbf{x}) \neq y_i\right]$

Generalization: Finite Case II

Almost always, prefect predictor does not exist: $\epsilon^\star = \min_i \mathrm{err}_D\big(f_i(x)\big) > 0$

Evaluate errors on S: $\mathrm{err}_S(f_i) = \epsilon_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[f_i(\mathbf{x}) \neq y_i\right]$

Choose f_j with the smallest empirical error: $\mathrm{err}_S(f_j)$

Generalization: Finite Case II

Almost always, prefect predictor does not exist: $\epsilon^\star = \min_i \mathrm{err}_D\big(f_i(x)\big) > 0$

Evaluate errors on S: $\mathrm{err}_S(f_i) = \epsilon_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[f_i(\mathbf{x}) \neq y_i\right]$

Choose f_j with the smallest empirical error: $\mathrm{err}_S(f_j)$

Generalization error of f_j is: $\mathrm{err}_D\big(f_j(x)\big) = \Delta\epsilon + \epsilon^\star = \Delta\epsilon + \min_i \mathrm{err}_D\big(f_i(x)\big)$

Generalization: Finite Case II

Almost always, prefect predictor does not exist: $\epsilon^\star = \min_i \mathrm{err}_D\big(f_i(x)\big) > 0$

Evaluate errors on S: $\mathrm{err}_S(f_i) = \epsilon_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\big[f_i(\mathbf{x}) \neq y_i\big]$

Choose f_j with the smallest empirical error: $\mathrm{err}_S(f_j)$

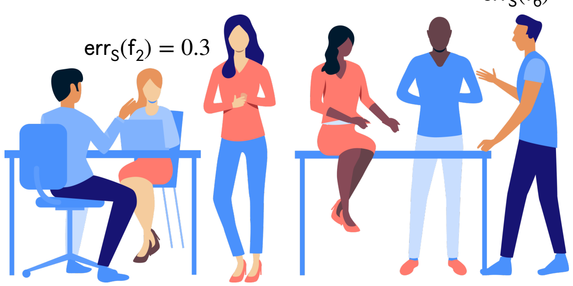
Generalization error of f_j is: $\mathrm{err}_D\big(f_j(x)\big) = \Delta\epsilon + \epsilon^\star = \Delta\epsilon + \min_i \mathrm{err}_D\big(f_i(x)\big)$

It takes $O\left(\frac{\log(k)}{(\Delta\epsilon)^2}\right)$ sample to get $\Delta\epsilon$-close to f of $\epsilon^\star = \min_i \mathrm{err}_D\big(f_i(x)\big) > 0$

$\mathrm{err}_S(f_4) = 0.16$

$\mathrm{err}_S(f_6) = 0.3$

$\mathrm{err}_S(f_2) = 0.3$

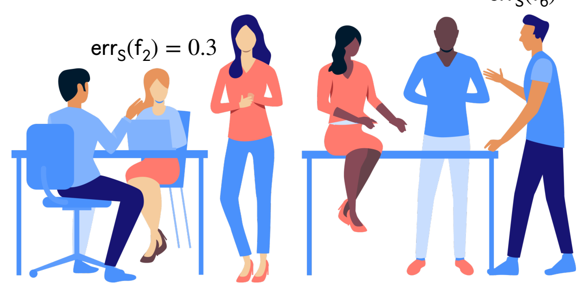


$\mathrm{err}_D(f_4) = 0.2$

$\mathrm{err}_S(f_4) = 0.16$

$\mathrm{err}_S(f_6) = 0.3$

$\mathrm{err}_S(f_2) = 0.3$



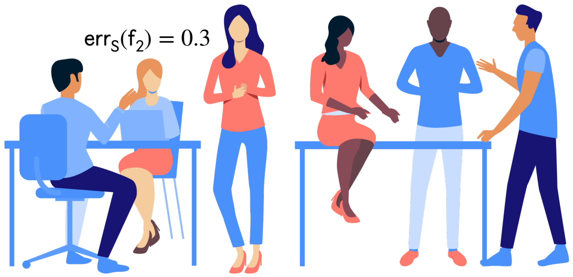
$\mathrm{err}_D(f_4) = 0.2$

$\Delta\epsilon = 0.04$

$\mathrm{err}_S(f_4) = 0.16$

$\mathrm{err}_S(f_6) = 0.3$

$\mathrm{err}_S(f_2) = 0.3$



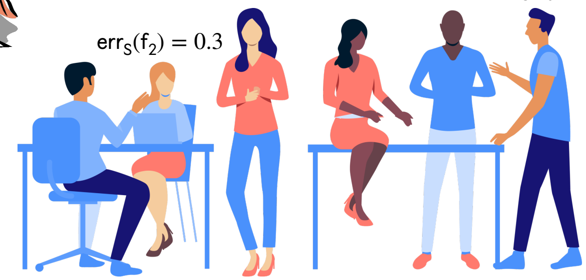
$\mathrm{err}_D(h^\star) = 0.14 \longleftrightarrow \mathrm{err}_D(f_4) = 0.2$

$\Delta\epsilon = 0.04$

$\mathrm{err}_S(f_4) = 0.16$

$\mathrm{err}_S(f_6) = 0.3$

$\mathrm{err}_S(f_2) = 0.3$



“Continuous Case”

“Continuous Case”

Find best model with weights $\mathbf{w} \in \mathbf{R}^d$

“Continuous Case”

Find best model with weights $\mathbf{w} \in \mathbf{R}^d$

For bfloat16: each entry of $\mathbf{w}$ can take 2^{16} different values

“Continuous Case”

Find best model with weights $\mathbf{w} \in \mathbf{R}^d$

For bfloat16: each entry of $\mathbf{w}$ can take 2^{16} different values

Each weight vector corresponds to bit vector of length16d

“Continuous Case”

Find best model with weights $\mathbf{w} \in \mathbf{R}^d$

For bfloat16: each entry of $\mathbf{w}$ can take 2^{16} different values

Each weight vector corresponds to bit vector of length16d

We have “only” 2^{16d} predictors $\Rightarrow f_1, \ldots, f_{2^{16d}}$

“Continuous Case”

Find best model with weights $\mathbf{w} \in \mathbf{R}^d$

For bfloat16: each entry of $\mathbf{w}$ can take 2^{16} different values

Each weight vector corresponds to bit vector of length16d

We have “only” 2^{16d} predictors $\Rightarrow f_1, \ldots, f_{2^{16d}}$

If $\exists \mathbf{w}^\star$ where $f_{\mathbf{w}^\star}(\mathbf{x}) = y$ for all $\mathbf{x}, y$ with $D(\mathbf{x}, y) > 0$ (0 generalization error):

it would take only $\tilde{O}(d)$ examples to find it !

"Continuous Case"

Find best model with weights $\mathbf{w} \in \mathbf{R}^d$

For bfloat16: each entry of $\mathbf{w}$ can take 2^{16} different values

Each weight vector corresponds to bit vector of length16d

We have "only" 2^{16d} predictors $\Rightarrow f_1, \ldots, f_{2^{16d}}$

If $\exists \mathbf{w}^\star$ where $f_{\mathbf{w}^\star}(\mathbf{x}) = y$ for all $\mathbf{x}, y$ with $D(\mathbf{x}, y) > 0$ (0 generalization error):

it would take only $\tilde{O}(d)$ examples to find it !

Caveats

Caveats

$\tilde{O}(d)$ hides pretty large constants

Caveats

$\tilde{O}(d)$ hides pretty large constants

Time of finding $\mathbf{w}^\star$ is exponential in **d**

Caveats

$\tilde{O}(d)$ hides pretty **large** constants

Time of finding $\mathbf{w}^\star$ is **exponential** in **d**

Not really learning — exhaustive search

Caveats

$\tilde{O}(d)$ hides pretty **large** constants

Time of finding $\mathbf{w}^\star$ is **exponential** in **d**

Not really learning — exhaustive search

Next step:

Incorporate mechanism called regularization into SGD