What would you most like to see covered on the review session?

“Review of CNN please!!”

“KL-divergence”

“regression penalties”

“Bias&Variance (the error decomposition is very confusing, the wiki page is clear and I thought I understand, but then a bit confused when the homework ask about a similar problem). Others are KL-divergence and boosting.”

“SVM”

“Regression, Bias-Variance Decomposition”

“RBFs, regression penalties,”

“Boosting, Decision trees construction (how to split in case of regression having ranges and how does it actually happen)”

“AdaBoost”

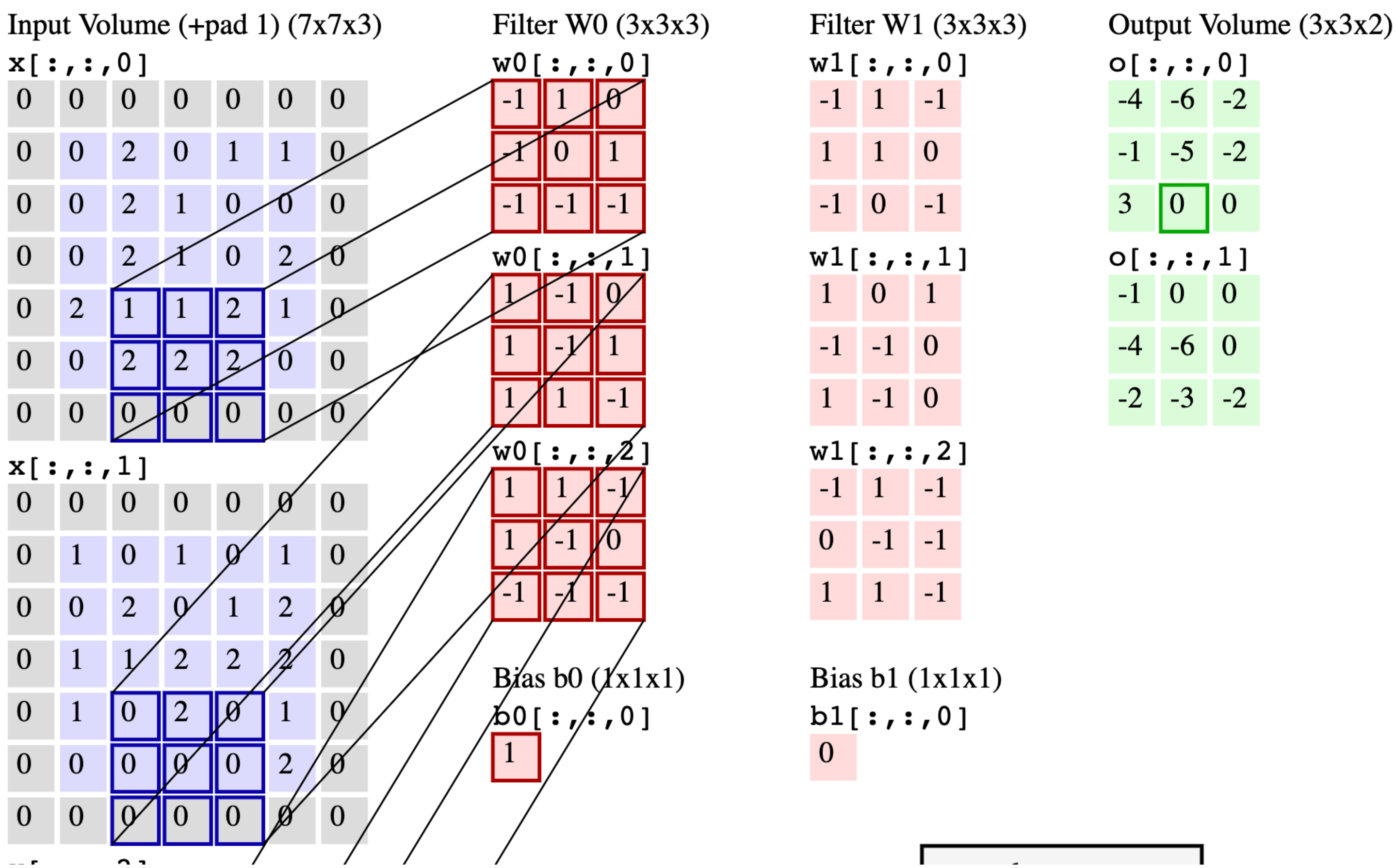
Announcements

- **Midterm Wednesday**
- **HW4 due Monday (no extensions)**
  - Solutions will be posted
- **Please don't cheat!**

Midterm Review

2020

CNN


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 2 0 1 1 0
0 0 2 1 0 0 0
0 0 2 1 0 2 0
0 2 1 1 2 1 0
0 0 2 2 2 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 0 1 0 1 0
0 0 2 0 1 2 0
0 1 1 2 2 2 0
0 1 0 2 0 1 0
0 0 0 0 0 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
-1 1 0
-1 0 1
-1 -1 -1
w0[:,:,1]
1 -1 0
1 -1 1
1 1 -1
w0[:,:,2]
1 1 -1
1 -1 0
-1 -1 -1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
-1 1 -1
1 1 0
-1 0 -1
w1[:,:,1]
1 0 1
-1 -1 0
1 -1 0
w1[:,:,2]
-1 1 -1
0 -1 -1
1 1 -1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
-4 -6 -2
-1 -5 -2
3 0 0
o[:,:,1]
-1 0 0
-4 -6 0
-2 -3 -2

Kullback Leibler divergence

- ***P*** = true distribution;
- ***Q*** = alternative distribution that is used to encode data
- **KL** divergence is the expected extra message length per datum that must be transmitted using ***Q***

$$D_{KL}(P \,||\, Q) = \Sigma_i P(x_i) \log (P(x_i)/Q(x_i))$$

$$= - \Sigma_i P(x_i) \log Q(x_i) + \Sigma_i P(x_i) \log P(x_i)$$

$$= \quad H(P,Q) \qquad\qquad - H(P)$$

= Cross-entropy – entropy

- **Measures how different the two distributions are**

Where do we use KL-divergence?

- $D(\, p(y \mid x, x') \;||\; p(\, y \mid x) \,)$
- $D(\, y \;||\; h(x) \,)$

Information and friends

- Entropy of the expected value of ____
- KL divergence is the expected value of ___
- Information gain is the difference between ___

Bias Variance Tradeoff

- **Bias:** if you estimate something many times (on different training sets, will you systematically be high or low?
- **Variance:** if you estimate something many times (on different training sets, how much does your estimate vary?

$$Bias(\hat{\theta}) = E[\hat{\theta} - \theta] = E[\hat{\theta}] - E[\theta]$$
$$Var(\hat{\theta}) = E[(\hat{\theta} - E[\hat{\theta}])^2]$$

Bias Variance Tradeoff - OLS

- Test Error = Variance + Bias^2 + Noise

$$\mathbf{E}_{x,y,D}[(h(x;D)-y)^2] = \underbrace{\mathbf{E}_{x,D}[(h(x;D)-\overline{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbf{E}_x[(\overline{h}(x)-\overline{y}(x))^2]}_{\text{Bias}^2} + \underbrace{\mathbf{E}_{x,y}[(\overline{y}(x)-y)^2]}_{\text{Noise}}$$

- This applies both to estimating w and to estimating y

$$\text{Error} = E[(y-\hat{y})^2] = Bias(\hat{y})^2 + Var(\hat{y}) + \sigma^2$$

Bias Variance Tradeoff - OLS

- Test Error = Variance + Bias^2 + Noise

$$\mathbf{E}_{x,y,D}[(h(x;D)-y)^2] = \underbrace{\mathbf{E}_{x,D}[(h(x;D)-\overline{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbf{E}_{x}[(\overline{h}(x)-\overline{y}(x))^2]}_{\text{Bias}^2} + \underbrace{\mathbf{E}_{x,y}[(\overline{y}(x)-y)^2]}_{\text{Noise}}$$

- This applies both to estimating w and to estimating y

$$\text{Error} = E[(y-\hat{y})^2] = Bias(\hat{y})^2 + Var(\hat{y}) + \sigma^2$$

Bias–Variance Trade-off

Higher complexity = larger or smaller?

bias^2	____	$\mathbf{E}_x[(\bar{h}(x) - \bar{y}(x))^2]$
variance	____	$\mathbf{E}_{x,D}[(h(x;D) - \bar{h}(x))^2]$
k of k-nn	____	
λ of L_p	____	
kernel width (RBF)	____	
? of decision trees	____	

Adaboost

Given: n examples $(\mathbf{x}_i, y_i)$, where $\mathbf{x} \in \mathcal{X}, y \in \pm 1$.

Initialize: $D_1(i) = \frac{1}{n}$

For $t = 1 \ldots T$

- Train weak classifier on distribution $D(i)$, $h_t(\mathbf{x}) : \mathcal{X} \mapsto \pm 1$
- Choose weight α_t (see how below)
- Update: $D_{t+1}(i) = \frac{D_t(i) \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}}{Z_t}$, for all i, where $Z_t = \sum_i D_t(i) \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}$

Output classifier: $h(\mathbf{x}) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

Where α_t is the log-odds of the weighted probability of the prediction being wrong

$$\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t} \qquad \epsilon_t = \sum_i D_t(i) \mathbf{1}(y_i \neq h_t(\mathbf{x_i}))$$

SVM: Hinge loss, ridge penalty

$h(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x} + b)$

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b))$$

0 if score is correct by 1 or more
(hinge loss)

SVM as constrained optimization

Hinge primal:

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b))$$

"Slack variable" – hinge loss from the margin

SVM dual

Hinge dual:

$$\max_{\alpha \geq 0} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \leq C, \quad i = 1, \ldots, n$$

$x_i^T x_j$ is the kernel matrix

C controls regularization

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

Scale invariance

- Decision tree?
- k-nn?
- OLS?
- Elastic net?
- L_0 penalized regression?
- SVM?

Kernel functions $k(\mathbf{x}_1,\mathbf{x}_2)$

- **Measure similarity or distance?**
- **How to check if something is a kernel function?**
  - Compute a Kernel matrix with elements $k(\mathbf{x}_i,\mathbf{x}_j)$
  - Make sure its eigenvalues are non-negative
- **Example:** $k(\mathbf{x}_i,\mathbf{x}_j) = x_{i1} + x_{i2} + x_{j1} + x_{j2}$
  - Try the single point $\mathbf{x} = (1,-2)$
  - $K(x,x) = 1-2+1-2 = [-3]$ which is a matrix with eigenvalue -3

Stepwise regression

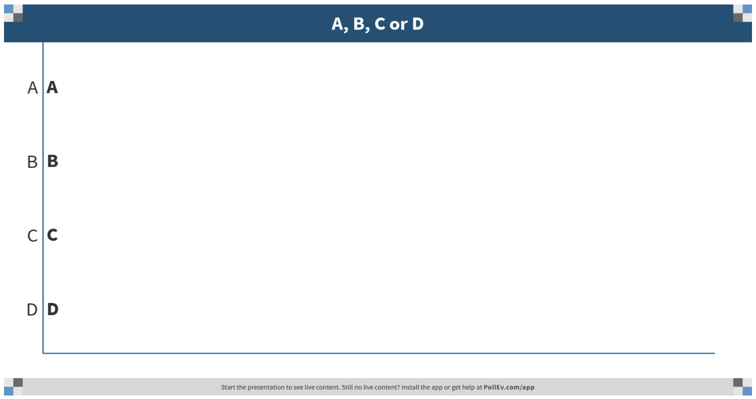
- ***Stepwise regression is used to minimize***

  A) Training set error (MLE)

  B) L_0 penalized training set error

  C) any penalized training set error

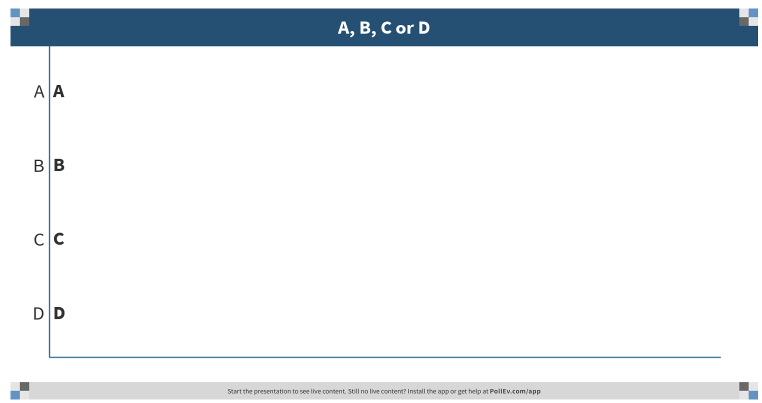
  D) None of the above



Why?

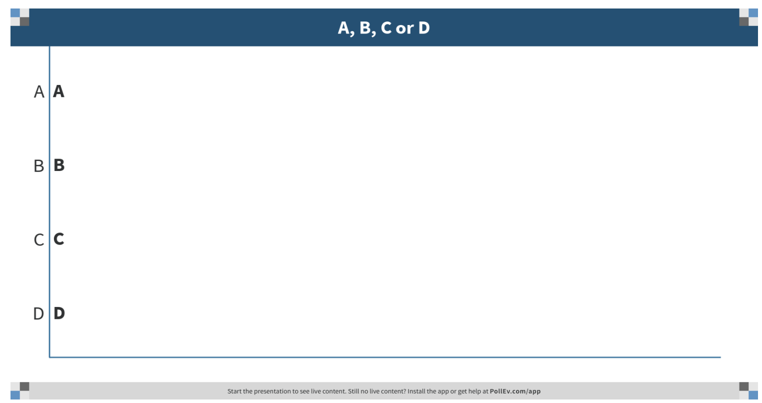
Stepwise regression

- **Given *p* features of which *q* end up being selected**
- ***Stepwise regression will estimate …***
  - *A) q regressions*
  - *B) p regressions*
  - *C) q p regressions*
  - *D) more regressions…*



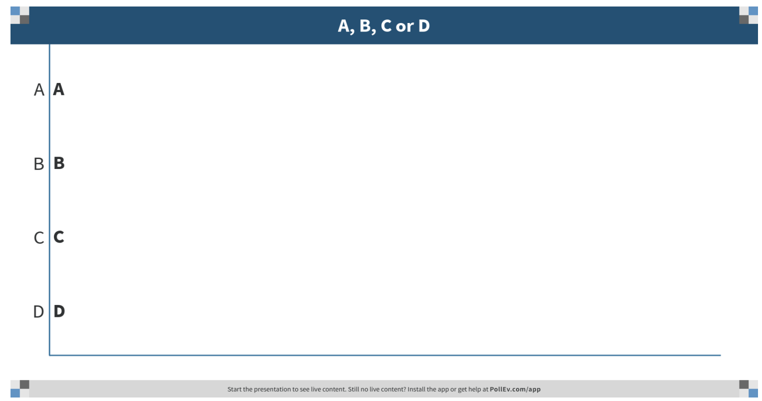
Streamwise regression

- **Given *p* features of which *q* end up being selected**
- ***Streamwise regression will estimate …***
  - *A) q regressions*
  - *B) p regressions*
  - *C) q p regressions*
  - *D) more regressions…*



Stagewise regression

- **Given *p* features of which *q* end up being selected**
- ***Stagewise regression will estimate …***
  - *A) q regressions*
  - *B) p regressions*
  - *C) q p regressions*
  - *D) more regressions…*



Stepwise regression

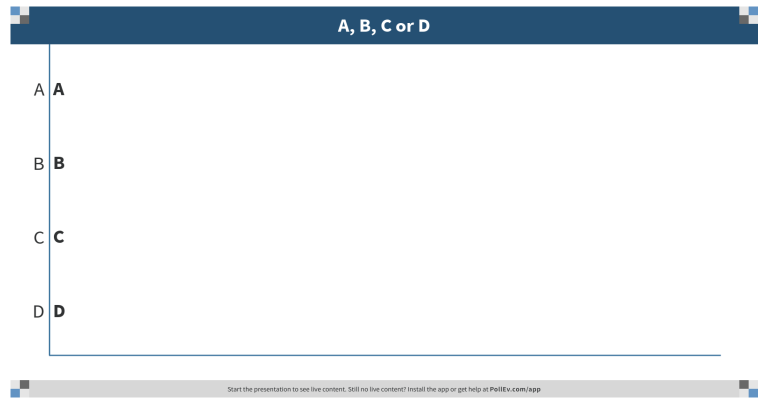
- Given *p* features of which *q* end up being selected
- *The largest matrix that needs to be inverted is*

  A) 1x1

  B) qxq

  C) pxp

  D) bigger



Stagewise regression

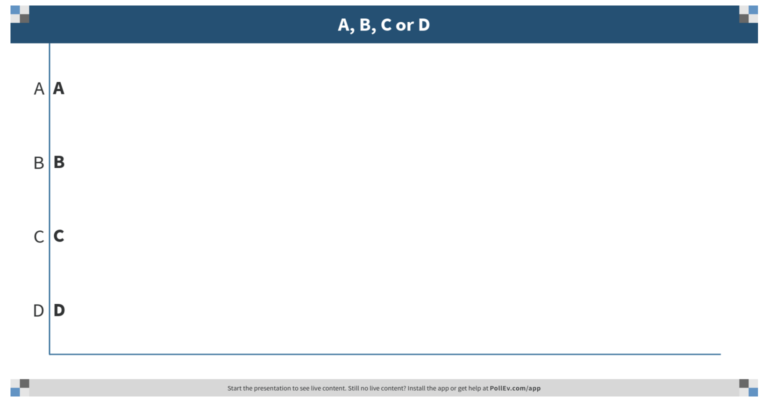
- **Given *p* features of which *q* end up being selected**
- ***The largest matrix that needs to be inverted is***

  A) 1x1

  B) qxq

  C) pxp

  D) bigger



Streamwise regression - example

- **Assume the true model is**

  $\mathbf{y = 2\,x_1 + 0\,x_2 + 2\,x_3 + 5\,x_4}$

  with $\mathbf{x_1 = x_3}$ (two columns are identical)

  and all features standardized

  – thus $\mathbf{x_4}$ will do the most to reduce the error

Streamwise: models are *y* =

$\mathbf{0,\ 4x_1,\ 4x_1,\ 4x_1\ ,\ 4x_1{+}5x_4}$

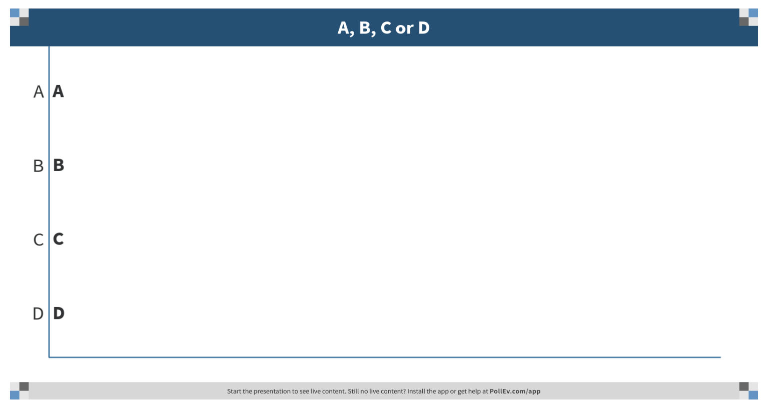
Stepwise: models are *y* =

$\mathbf{0,\ \ 5x_4,\ 4x_1{+}5x_4}$ *or* $\mathbf{4x_3{+}5x_4}$

RBF

- **Transform X to Z using**
  - $z_{ij} = \phi_j(x_i) = k(x_i, \mu_j)$
  - How many μ_j do we use?
    - A) $k < p$
    - B) $k = p$
    - C) $k > p$
    - D) any of the above
  - How do we pick k?
  - What other complexity tuner do we have?
- **Linearly regress y on Z**

RBF uses what kernel?



$$y_i = \Sigma_j \, a_j \, \phi_j(\mathbf{x}_i)$$

Kernel question

x	y
(1,1)	+1
(1,0)	-1
(0,1)	-1
(-1,1)	+1

Is this linearly separable?

Can you make this linearly separable with 4 Gaussian kernels?

Can you make this linearly separable with 2 Gaussian kernels?

Can you make this linearly separable with 1 Gaussian kernel?

Logistic Regression

$$P(Y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1+\exp\{-\sum_j w_j x_j\}} = \frac{1}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{1}{1+\exp\{-y\mathbf{w}^\top \mathbf{x}\}}$$

$$P(Y = -1|\mathbf{x}, \mathbf{w}) = 1 - P(Y = 1|\mathbf{x}, \mathbf{w}) = \frac{\exp\{-\mathbf{w}^\top \mathbf{x}\}}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{1}{1+\exp\{-y\mathbf{w}^\top \mathbf{x}\}}$$

$$log(\frac{P(Y=1|\mathbf{x},\mathbf{w})}{P(Y=-1|\mathbf{x},\mathbf{w})}) = \mathbf{w}^\top \mathbf{x}$$

Log likelihood of data

$$\log(P(D_Y|D_X, \mathbf{w})) = \log\left(\prod_i \frac{1}{1+\exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}} \right)$$

$$= -\sum_i \log(1 + \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\})$$

Decision Boundary

$$P(Y = 1|\mathbf{x}, \mathbf{w}) = P(Y = -1|\mathbf{x}, \mathbf{w})$$

$$\frac{1}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{\exp\{-\mathbf{w}^\top \mathbf{x}\}}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}}$$

$$\mathbf{w}^\top \mathbf{x} = 0$$

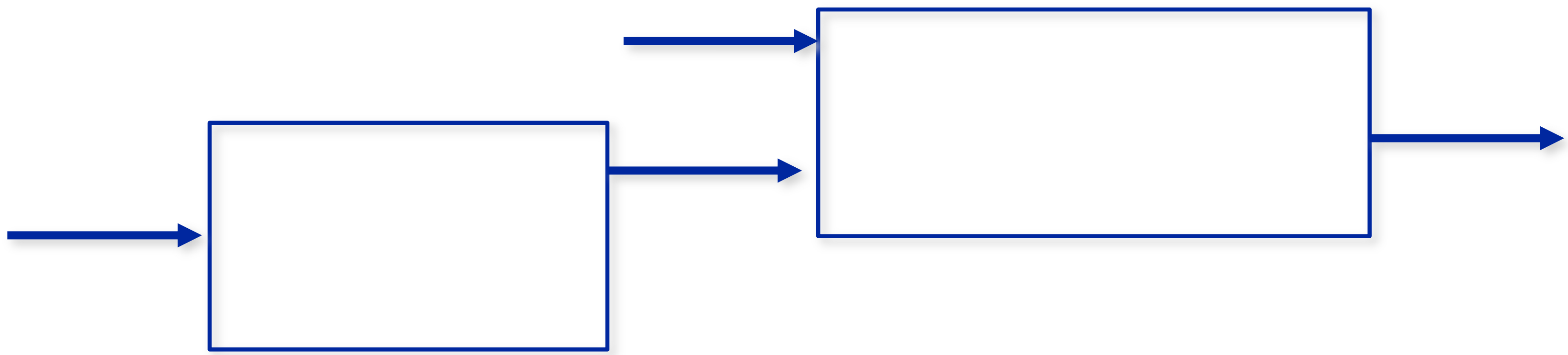
Prediction: $y = \text{sign}(\mathbf{w}^\top \mathbf{x})$

k-class logistic regression

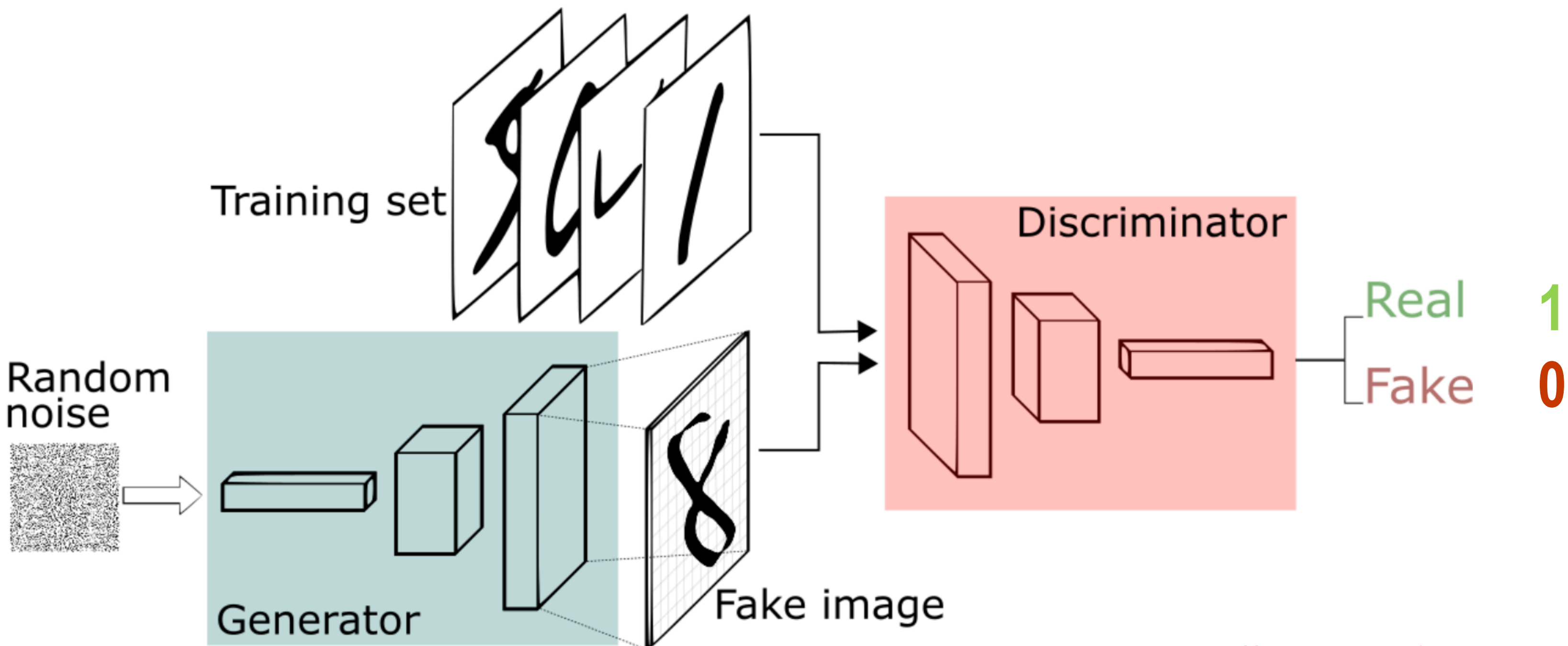
$$P(Y = k|\mathbf{x}, \mathbf{w}) = \frac{\exp\{\mathbf{w}_k^\top \mathbf{x}\}}{\sum_{k'=1}^{K} \exp\{\mathbf{w}_{k'}^\top \mathbf{x}\}}, \quad \text{for} \quad k = 1, \ldots, K$$

Prediction: $y = \mathrm{argmax}_k(\mathbf{w}_k^T \mathbf{x})$

GANS



Generative Adversarial Networks: GANS



G: argmin log(1 − D(G(noise)))

https://medium.freecodecamp.org/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394

Conditional GANS