ML PR 1 7, 2018

1. Recursive feature elimination

Progressively remove features that seem to improve performance the most

- ~~[illegible]~~ Backward elimination starts with full set and removes variables one by one

☐ Kernel trick

SVM can be extended to non linear boundaries using the kernel trick that maps the data into a higher dimension and redesigns the SVM

- Map the 2D samples into 3D : $(x, y) \to (x^2, y^2, \sqrt{2}\, xy)$
- Train the SVM with the new 3D samples
- The decision boundary is linear in 3D but non linear in 2D

☐ Likelihood function (likelihood = probability)

It is a function of the parameters of a statistical problem given specific observed data

2. $x[n] = A\,s[n] + w[n] \quad n = 0, \dots, N-1$

$w[n] \sim N(0, \sigma^2)$ with $s[n]$ as a known signal

$$P(x; A) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left[-\frac{1}{2\sigma^2} \sum_{n=0}^{N-1} (\underbrace{x[n] - A\,s[n]}_{w[n]})^2 \right]$$

$$\ln P(X; A) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=0}^{N-1} \underbrace{(x[n] - A\,s[n])^2}_{x^2[n] + A^2 s^2[n] - 2x[n]A\,s[n]}$$

$$\frac{\partial}{\partial A} \ln P(X; A) = -\frac{1}{2\sigma^2} \sum_{n=0}^{N-1} (2A\,s^2[n] - 2x[n]\,s[n]) = 0$$

$$A = \frac{\sum_{n=0}^{N-1} x[n]\,s[n]}{\sum_{n=0}^{N-1} s^2[n]}$$

Scanned by CamScanner

ML and PR exam 25.4.2016

2. □ Likelihood ratio test

$$H_1: x[n] = A + w[n] \rightarrow \text{signal + noise}$$
$$H_0: x[n] = w[n] \rightarrow \text{noise}$$
$$w[n] \sim N(0, \sigma^2)$$

~~Pharm~~ Probabilities of both hypotheses are compared, and the one with signal + noise has greater value. Then, they are compared in a ratio which is function of a threshold γ

□ K-nearest neighbor classifier

The approach is based on seeing what kind of samples are nearby, that is, to copy the class label of the most similar training sample to the unknow test sample.
Even though, it is fragile to changes in the training data. That is, the classification boundary may change a lot by moving only one training sample.
The robustness is increased by taking the majority vote of more nearby samples. It selects the most frequent class label among the K nearest training samples.

□ Cross-validation
The generalization ability of a classifier needs to be tested without seen samples. This can be done by splitting the data into separate training and test sets.
A standard approach is to use K-fold cross-validation:
- Split the training data to K parts (folds)
- Use each fold for testing exactly once and train with other folds
- The error estimate is the mean of the K estimates

$K \in [5, 10]$

Scanned by CamScanner

□ Convolutional neural network

It is an architecture that preserves the topology of the input

The structure is : $\boxed{\text{convolution}} \to \boxed{\text{non-linearity}} \to \boxed{\text{subsampling}}$

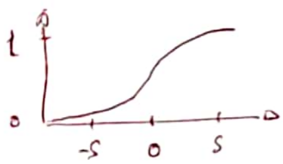
- Convolution : filters the input with a number of convolutional kernels
  the filters see local window from all RGB layers
  the results are the feature maps (typically dozens)

- ReLU : passes the feature maps through a pixelwise Rectified Linear Unit
  $ReLU(x) = \max(0, x)$

- Subsampling : it shrinks the input dimensions by an integer factor
  Maxpooling (takes max of each 2x2 block)
  It reduces data size and improves spacial invariance

□ Logistic function

Also known as sigmoid activation in neural networks context



It is used in logistic regression classifiers and it is based

$$f(x) = \frac{1}{1 + \exp[-(w^T x + b)]}$$

It maps the projection $w^T x + b$ limiting it to the range $[0, 1]$

B. b The projected Gaussians are univariate normal:

$N(w^T\mu_1, w^T C_1 w)$ and $N(w^T\mu_2, w^T C_2 w)$

Formulate classification problem as likelihood ratio test and choose threshold

$$N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$H_1: P(x \mid \mu = w^T\mu_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)$$

$$H_2: P(x \mid \mu = w^T\mu_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)$$

$$H_1 > \gamma H_2 \to \frac{H_1}{H_2} > \gamma$$

$$\frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2} - \left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)\right] > \gamma$$

$$-\frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2} > \ln \gamma \frac{\sqrt{\sigma_2^2}}{\sqrt{\sigma_1^2}}$$

$$2\sigma_1^2\sigma_2^2 \ln \gamma \sqrt{\frac{\sigma_2^2}{\sigma_1^2}} + (x-\mu_1)^2\sigma_2^2 - (x-\mu_2)^2\sigma_1^2 < 0$$

4. Compute gradient for L_2 penalized log-loss

$$\ell(w) = \underbrace{\sum_{n=0}^{N-1} \ln(1+\exp(y_n w^T x_n))}_{a} + \underbrace{\lambda w^T w}_{b}$$

$$\frac{\partial}{\partial w}(\lambda w^T w) = \lambda 2 w^T = 2\lambda w^T$$

$$\frac{\partial}{\partial w} \sum_{n=0}^{N-1} \ln(1+\exp(y_n w^T x_n)) = \sum_{n=0}^{N-1} \frac{1}{(1+\exp(y_n w^T x_n))} \frac{\partial}{\partial w}(1+\exp(y_n w^T x_n))$$

$$= \sum_{n=0}^{N-1} \frac{\exp(y_n w^T x_n)}{1+\exp(y_n w^T x_n)} \frac{\partial}{\partial w}(y_n w^T x_n) = \sum_{n=0}^{N-1} \frac{y_n x_n \exp(y_n w^T x_n)}{1+\exp(y_n w^T x_n)}$$

$$\frac{\partial}{\partial w} \ell(w) = \sum_{n=0}^{N-1} \frac{y_n x_n \exp(y_n w^T x_n)}{1+\exp(y_n w^T x_n)} + 2\lambda w^T$$

Exam PR ML 1.3.2017

1. Define

☐ Rectified Linear unit

It is an activation function used in neural networks

$f(x) = x^+ = \max(0, x)$ where x is the input of a neuron

☐ linear classifier

It learns a linear decision boundary between classes

$$F(x) = \begin{cases} \text{Class } 1, & \text{if } w^T x < b \\ \text{Class } 2, & \text{if } w^T x \geq b \end{cases}$$

where b is the threshold and w the learned weights. x is the data.

LDA, SVM, logistic Regression

☐ Ensemble classifier

They are composed by a group of individual weak classifiers

Non-linear

Random Forests, Ada Boost paradigm, Gradient Boosted regression trees,

Extremely randomized trees, ...

☐ Multilabel classifier

If the number of classes are non-binary, that is, there are more than two classes, then the classifier is multidimensional or multilabel.

☐ Stratified cross-validation

It gives better results than completely random split. Each fold will hold a mixture of classes in same proportion as in full dataset

☐ L_1 regularization

Adds a penalty term to the fitting error so the model is encouraged to use small coefficients because large ones are expensive

In regression: $y_n = w^T x_n + e_n$ where $e_n \sim N(0, \sigma^2)$

$$\underset{w}{\text{minimize}} \left(\sum_{n=0}^{N-1} (y_n - w^T x_n)^2 \right) \longrightarrow w_{LS} = (X^T X)^{-1} X^T y$$

L_1 : $\underset{w}{\text{minimize}} \sum_{n=0}^{N-1} \left[(y_n - w^T x_n)^2 + \lambda \|w\|_1 \right]$ $\quad \|w_1\| = \sum_{p=1}^{P} |w_p|$

LASSO

With this penalty there is no close form expression

The minimum is solved iteratively using, e.g., gradient search

L_2 : $\underset{w}{\text{minimize}} \sum_{n=0}^{N-1} \left[(y_n - w^T x_n)^2 + \lambda w^T w \right]$

$$w_{RIDGE} = (X^T X + \lambda I)^{-1} X^T y$$

4. $\ell(w) = \sum_{n=0}^{N-1} \ln(1 + \exp(y_n w^T x_n))$

$$\frac{\partial}{\partial w} \ell(w) = \sum_{n=0}^{N-1} \frac{1}{1 + \exp(y_n w^T x_n)} \frac{\partial}{\partial w} (\overset{0}{\cancel{1}} + \exp(y_n w^T x_n)) =$$

$$= \sum_{n=0}^{N-1} \frac{\exp(y_n w^T x_n)}{1 + \exp(y_n w^T x_n)} \underbrace{\frac{\partial}{\partial w} (y_n w^T x_n)}_{y_n x_n} = \sum_{n=0}^{N-1} \frac{y_n x_n \exp(y_n w^T x_n)}{1 + \exp(y_n w^T x_n)}$$

Stochastic gradient. Compute next iteration for w when $x_n = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, y_n = 1$ and $w = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

$$w_{n+1} = w_n - \eta \frac{\partial \ell(w)}{\partial w} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \eta \frac{1 \begin{bmatrix} -1 \\ 1 \end{bmatrix} \exp\left(1 \cdot [2, 1] \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right)}{1 + \exp\left(1 \cdot [2 \ 1] \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right)} =$$

$$= \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \eta \frac{\begin{bmatrix} -1 \\ 1 \end{bmatrix} \exp(-1)}{1 + \exp(-1)} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \eta \frac{e^{-1}}{1 + e^{-1}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

ML PR Scripts in exams

- 1. 3. 2017

~~X_train, ...~~

```
X_train, y_train = load_training_data()
X_test, y_test = load_test_data()
# list
classifiers = [(LogisticRegression(), "Logistic Regression"),
               (SVC(), "Support Vector Machine"),
               (RandomForestClassifier(), "Random Forest")]

for clf, name in classifiers:

    accuracies = []
    for iteration in range(len(y_train))
        clf.fit(X_train, y_train)
        y_hat = clf.predict(X_test)
        accuracy = accuracy_score(y_test, y_hat)
        score.append(accuracy)

    print("Accuracy: %.2f" % (score))
```

PR ML Exam 27.2.2018

1. □ Least Squares estimator minimizes the squared distance between all samples

□ The Receiver Operating Characteristics curve plots the probability of detection versus the probability of false alarm for all thresholds

□ The LDA maximizes $J(w) = \frac{\text{Distance of class means}}{\text{Variance of classes}}$

□ A NN classifier has a non-linear decision boundary between classes

□ L1 and L2 regularization improves the generalization of a logistic regression classifier

□ Max pooling does not returns the maximum over input channels but over window size

2.

n	0	1	2
$x(n)$	7	9	2
$y(n)$	11.6	14.8	3.5

$y(n) = ax(n) + b$

Find L_2-regularized least squares estimates $\hat{a}$ and $\hat{b}$ that minimize the squared error using penalty $\lambda = 10$

Least Squares model : $\hat{\theta} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = (X^T X)^{-1} X^T y$

where $X = \begin{bmatrix} 7 & 1 \\ 9 & 1 \\ 2 & 1 \end{bmatrix}$ is the Vandermode matrix

Last column all ones

Regularized Least Squares model : $\hat{\theta} = w_{RIDGE} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = (X^T X + \lambda I)^{-1} X^T y$

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \left(\begin{bmatrix} 7 & 9 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 & 1 \\ 9 & 1 \\ 2 & 1 \end{bmatrix} + 10 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 7 & 9 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 11.6 \\ 14.8 \\ 3.5 \end{bmatrix} =$$

$$= \Bigg(\overbrace{\begin{bmatrix} 134 & 18 \\ 18 & 3 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}}^{\begin{bmatrix} 144 & 18 \\ 18 & 13 \end{bmatrix}} \Bigg)^{-1} \begin{bmatrix} 7 & 9 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 11.6 \\ 14.8 \\ 3.5 \end{bmatrix} = \begin{bmatrix} 1.51 \\ 0.21 \end{bmatrix}$$

$$\underbrace{\qquad}_{\begin{bmatrix} 0.0084 & -0.01163 \\ -0.01163 & 0.093 \end{bmatrix}}$$

3. LDA find weight vector and threshold c at the center of the projected class means

$$\text{Class}(x) = \begin{cases} 1, & \text{if } w^T x \geq c \\ 2, & \text{otherwise} \end{cases}$$

$$C_1 = \frac{1}{3}\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1/3 & 1/3 \\ 1/3 & 2/3 \end{pmatrix} \qquad C_2 = \frac{1}{3}\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 5/3 & 1/3 \\ 1/3 & 1/3 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} -1.5 \\ -1 \end{pmatrix} \qquad \mu_2 = \begin{pmatrix} 1.2 \\ -3/2 \end{pmatrix}$$

$$w = S_w^{-1}(\mu_1 - \mu_0) = (C_1 + C_2)^{-1}(\mu_2 - \mu_1) =$$

$$= \left(\frac{1}{3}\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 5 & 1 \\ 1 & 1 \end{bmatrix}\right)^{-1}\left(\begin{bmatrix} 1.2 \\ -3/2 \end{bmatrix} - \begin{bmatrix} -1.5 \\ -1 \end{bmatrix}\right) =$$

$$= \begin{bmatrix} 2 & 2/3 \\ 2/3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2.7 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 1.95 \\ -1.8 \end{bmatrix}$$

$$\cancel{c = \frac{[1.95 \quad -1.8]\begin{bmatrix} -1.5 \\ -1 \end{bmatrix} + [1.95 \quad -1.8]\begin{bmatrix} 1.2 \\ -3.2 \end{bmatrix}}{2} = 3.4875}$$

~~T = np.mean([np.dot(m1, w), np.dot(m2, w)])~~

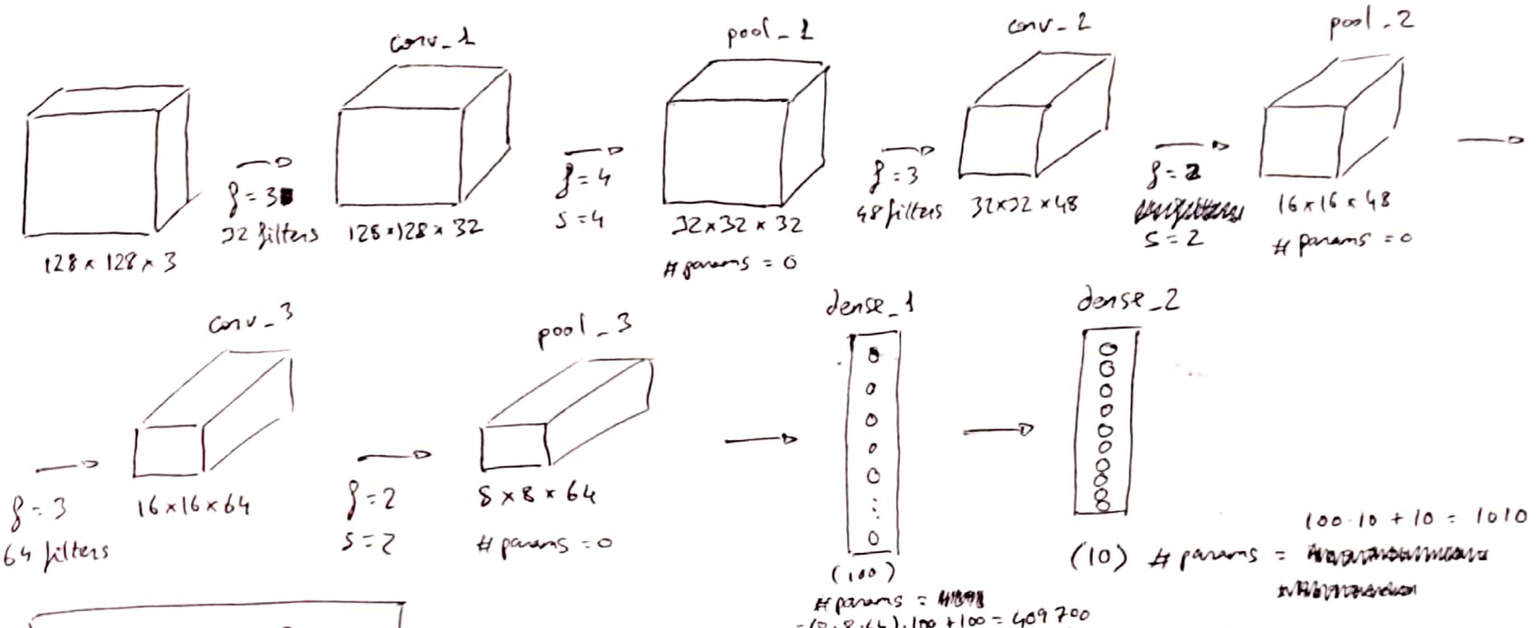
$$T = \frac{1}{2} w^T(\mu_1 + \mu_2) = \frac{1}{2}[1.95 \quad -1.8]\left(\begin{bmatrix} -1.5 \\ -1 \end{bmatrix} + \begin{bmatrix} 1.2 \\ -3/2 \end{bmatrix}\right) =$$

$$= \frac{1}{2}[1.95 \quad -1.8]\begin{bmatrix} -0.3 \\ -5/2 \end{bmatrix} = 1.9575$$

4. Inputs 128 × 128 color images from 10 categories

a) Draw a diagram of the network given by model.summary()

b) Compute # parameters of each layer and their total number over all layers



$$\boxed{\gamma = \frac{n + 2p - f}{s} + 1}$$

padding = "same" → output γ = input n

$$\gamma_{pool\_1} = \frac{128 + 2\cdot 0 - 4}{4} + 1 = 32 \qquad \gamma_{pool\_2} = \frac{32 + 2\cdot 0 - 2}{2} + 1 = 16$$

$$\gamma_{pool\_3} = \frac{16 + 2\cdot 0 - 2}{2} + 1 = 8$$

$$\boxed{\#\,params\_conv = f^2\, c\,(\#filters) + (\#filters)}$$

} 453190 params in total

$\#\,params\_conv\_1 = 3^2\cdot 3(32) + (32) = 896$

$\#\,params\_conv\_2 = 3^2\, 32\cdot 48 + 48 = 13872$

$\#\,params\_conv\_3 = 3^2\cdot 48\cdot 64 + 64 = 27712$

PR ML Exam 10.4.2018

1.

- ☐ Maximum likelihood estimators are biased or unbiased
- ☐ Least squares estimator minimizes the squared distances between the data and the model
- ☐ The number of support vectors of a support vector machine is ~~[illegible]~~ different of the total number of samples
- ☐ Random forest has a non-linear decision boundary
- ☐ Stratified cross-validation resamples the training data such that all classes have equal number of samples.

2. Poisson distribution → discrete probability distribution which describes probability of a number of events $x = 0, 1, 2, \dots$ occurring in a fixed period of time

$$P(x;\lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!}, \quad \lambda > 0 \text{ defining the shape of the density}$$

$x[n]$ with $n = 0, 1, \dots, N-1$

a) Find maximum likelihood estimator of λ

$$P(x;\lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!} \longrightarrow P(x;\lambda) = \sum_{n=0}^{N-1} \frac{e^{-\lambda}\lambda^{x[n]}}{x[n]!}$$

$$\ln P(x;\lambda) = \sum_{n=0}^{N-1}\left[\ln e^{-\lambda} + \ln \lambda^{x[n]} - \ln(x[n])!\right] =$$

$$= \underbrace{\sum_{n=0}^{N-1}}_{-N\lambda}\left[-\lambda + x[n]\ln\lambda - \ln(x[n])!\right]$$

$$\frac{\partial}{\partial\lambda}\ln P(x;\lambda) = -N + \sum_{n=0}^{N-1} x[n]\frac{1}{\lambda} - 0 = 0 \longrightarrow \lambda = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

↳ equal to the mean

↳ unbiased

3. Gaussian distribution of two datasets

$$\mu_0 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \qquad \mu_1 = \begin{bmatrix} 0 \\ -3 \end{bmatrix}$$

$$C_0 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \qquad C_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Calculate LDA projection vector w

- Solution without Eigen values:

$$w = S_w^{-1}(\mu_1 - \mu_0) = (C_0 + C_1)^{-1}(\mu_1 - \mu_0) =$$

$$= \left(\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} 0 \\ -3 \end{bmatrix} - \begin{bmatrix} 3 \\ 1 \end{bmatrix} \right) =$$

$$= \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}^{-1} \begin{bmatrix} -3 \\ -2 \end{bmatrix} = \frac{1}{\underbrace{4 \cdot 3 - 1 \cdot 1}_{1/11}} \begin{bmatrix} 3 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} -3 \\ -2 \end{bmatrix} =$$

$$= \begin{bmatrix} -0.\overline{63} \\ -0.\overline{45} \end{bmatrix}$$

$$P_D = P_{\text{True Positive Rate}} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

5.

$$P_{FA} = P_{\text{False Positive Rate}} = \frac{\text{False positive}}{\text{False positives} + \text{True negatives}}$$

Sample	Prediction	True label
1	0.8	1
2	0.5	1
3	0.6	0
4	0.4	0
5	0.2	0

		Actual: Detection	Actual: No detection
Predicted	Detection	True positive	False positive
	No detection	False negative	True negative

(1) $T < 0.2 \rightarrow P_D = 1.0 \quad P_{FA} = 1.0$ $\qquad$ (1) $\begin{bmatrix} 2 & 3 \\ 0 & 0 \end{bmatrix}$

(2) $0.2 \le T < 0.4 \rightarrow P_D = 1.0 \quad P_{FA} = 0.67$ $\qquad$ (2) $\begin{bmatrix} 2 & 2 \\ 0 & 1 \end{bmatrix}$

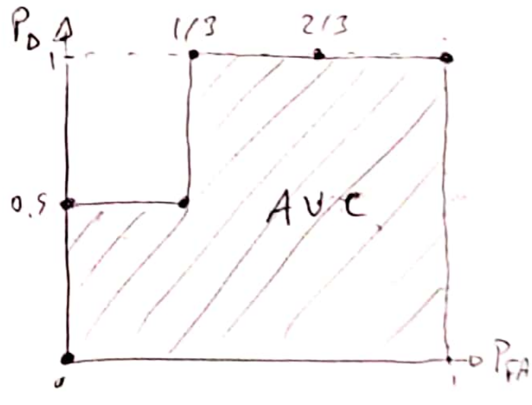
(3) $0.4 \le T < 0.5 \rightarrow P_D = 1.0 \quad P_{FA} = 0.33$ $\qquad$ (3) $\begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$

(4) $0.5 \le T < 0.6 \rightarrow P_D = 0.5 \quad P_{FA} = 0.33$ $\qquad$ (4) $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$

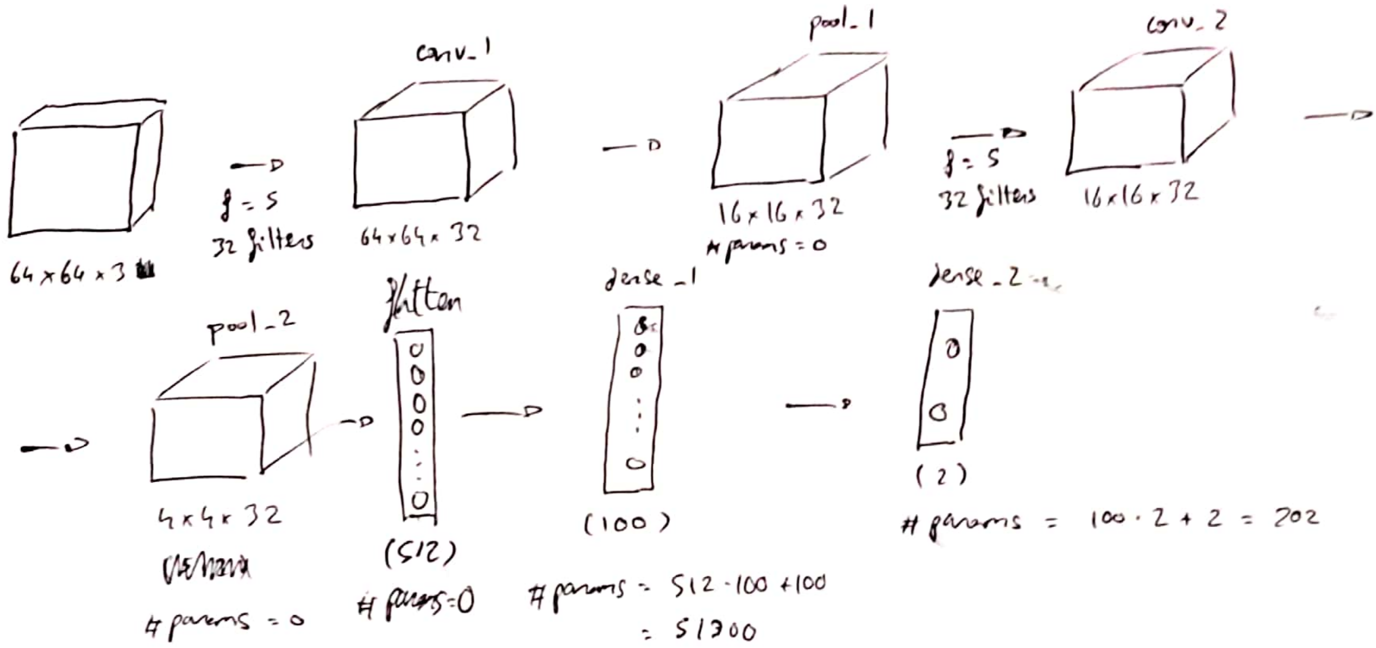
(5) $0.6 \le T < 0.8 \rightarrow P_D = 0.5 \quad P_{FA} = 0.0$ $\qquad$ (5) $\begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix}$

(6) $T > 0.8 \rightarrow P_D = 0.0 \quad P_{FA} = 0.0$ $\qquad$ (6) $\begin{bmatrix} 0 & 0 \\ 2 & 3 \end{bmatrix}$

$$AUC = \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0.5 = 5/6 = 0.8\overline{3}$$



4.



$$\# \text{ conv\_params} = f^2 c\,(\# \text{filters}) + (\# \text{filters})$$

$$\# \text{ conv\_params\_1} = 5^2 \cdot 3\,(32) + 32 = 2432$$

$$\# \text{ conv\_params\_2} = 5^2 \cdot 32 \cdot 32 + 32 = 25632$$

c) How many scalar multiplications take place on the first convolutional layer?

$$5^2 \cdot 3 \cdot 32 = 2400$$

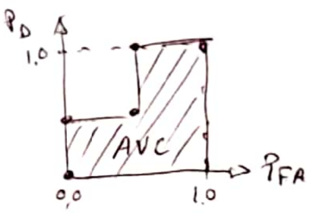
5. Draw ROC and guess AUC

$\begin{array}{|c|c|} \hline T+ & F+ \\ \hline F- & T- \\ \hline \end{array}$

		Actual: Detection	Actual: No detection
Prediction	Detection	True positive	~~False negative~~ False positive
	No detection	False negative	True negative

Sample	Prediction	True label
1	0.8	1
2	0.3	1
3	0.5	0
4	0.25	0

$$P_D = \frac{(T+)}{(T+)+(F-)} \qquad P_{FA} = \frac{(F+)}{(F+)+(T-)}$$



$$AUC = \frac{3}{4}$$

(1) $T < 0.25$: $P_D = 1.0$ $P_{FA} = 1.0$

(2) $0.25 \le T < 0.3$: $P_D = 1.0$ $P_{FA} = 0.5$

(3) $0.3 \le T < 0.5$: $P_D = 0.5$ $P_{FA} = 0.5$

(4) $0.5 \le T < 0.8$: $P_D = 0.5$ $P_{FA} = 0$

(5) $T \ge 0.8$: $P_D = 0$ $P_{FA} = 0$

(1) $\begin{array}{|c|c|} \hline 2 & 2 \\ \hline 0 & 0 \\ \hline \end{array}$ (2) $\begin{array}{|c|c|} \hline 2 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$ (3) $\begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$ (4) $\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 1 & 2 \\ \hline \end{array}$ (5) $\begin{array}{|c|c|} \hline 0 & 0 \\ \hline 2 & 2 \\ \hline \end{array}$

$-\frac{x^2[n]\, 2\sigma}{(2\sigma)^2} =$

PR ML Exam 11.4.2017

1. ☐ logistic regression classifier has ~~nonlinear~~ NON linear decision boundary between classes

☐ Dropout regularization improves the generalization of a neural network

☐ Stratified cross-validation resamples the training data such that all classes have equal number of samples

2. The Rayleigh distribution is a probability distribution used e.g. when modeling magnitude of a vector field.

$$p(x;\sigma) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right) \quad \text{for } x > 0$$

We measure N samples: $x_0, \dots, x_{N-1}$

a) Compute probability $p(x;\sigma)$ of observing the samples $x = (x_0, \dots, x_{N-1})$

b) " Logarithm of $p(X;\sigma)$ and differenciate result with respect σ

c) Find the maximum

$$p(x;\sigma) = \sum_{n=0}^{N-1}\left[\frac{x[n]}{\sigma^2}\exp\left(-\frac{x^2[n]}{2\sigma^2}\right)\right]$$

$$\ln p(x;\sigma) = \ln\left(\sum_{n=0}^{N-1}\left[\frac{x[n]}{\sigma^2}\exp\left(-\frac{x^2[n]}{2\sigma^2}\right)\right]\right) = \sum_{n=0}^{N-1}\ln\frac{x[n]}{\sigma^2} + \sum_{n=0}^{N-1}\ln\left(\exp\left(-\frac{x^2[n]}{2\sigma^2}\right)\right) =$$

$$= \sum_{n=0}^{N-1}\ln\frac{x[n]}{\sigma^2} - \sum_{n=0}^{N-1}\frac{x^2[n]}{2\sigma^2} = \sum_{n=0}^{N-1}\ln x[n] - \sum_{n=0}^{N-1}\ln\sigma^2 - \sum_{n=0}^{N-1}\frac{x^2[n]}{2\sigma^2}$$

$$\frac{x^2[n]\,\sigma^{-2}}{2}$$

$$\frac{\partial}{\partial\sigma}\ln p(x;\sigma) = -\sum_{n=0}^{N-1}\frac{1}{\sigma^2}2\sigma - \sum_{n=0}^{N-1}\frac{-2x^2[n]\sigma^{-3}}{2} =$$

$$\frac{2}{\sigma}$$

$$= -\sum_{n=0}^{N-1}\frac{2}{\sigma} + \sum_{n=0}^{N-1}\frac{x^2[n]}{\sigma^3} = \sum_{n=0}^{N-1}\left(\frac{-2}{\sigma} + \frac{x^2[n]}{\sigma^3}\right) = \frac{1}{\sigma}\sum_{n=0}^{N-1}\left(-2 + \frac{x^2[n]}{\sigma^2}\right) \equiv 0$$

$$\sigma^2 = \sum_{n=0}^{N-1}\frac{x^2[n]}{2} \rightarrow \sigma = \sum_{n=0}^{N-1}\frac{x[n]}{\sqrt{2}}$$

Scanned by CamScanner

~~3. a) Find one linear classifier (except LDA)~~

~~$Class(x) = \begin{cases} 1, & \text{if} \\ 2, & \text{otherwise} \end{cases}$~~

5. b) In lectures → kernel trick in support vectors $K(x,y) = (x \cdot y)^2$ for $x = (x_1, x_2)$ and $y = (y_1, y_2)$ corresponds to the mapping

$$\begin{pmatrix} u \\ v \end{pmatrix} \to \begin{pmatrix} u^2 \\ v^2 \\ \sqrt{2}uv \end{pmatrix}$$

if $K(x,y) = x \cdot y$ → linear kernel

$$K(x,y) = (x^T y)^2 = (x_1 y_1 + x_2 y_2)^2 = (x_1 y_1)^2 + (x_2 y_2)^2 + 2 x_1 y_1 x_2 y_2 =$$

$$= (x_1 y_1)^2 + (x_2 y_2)^2 + (\sqrt{2}\, x_1 x_2)(\sqrt{2}\, y_1 y_2) =$$

$$= (x_1^2, x_2^2, \sqrt{2} x_1 x_2) \cdot (y_1^2, y_2^2, \sqrt{2} y_1 y_2)$$

$$\begin{pmatrix} u \\ v \end{pmatrix} \to \begin{pmatrix} u^2 \\ v^2 \\ \sqrt{2}uv \end{pmatrix}$$

$$K(x,y) = (x^T y + 1)^2 = (x_1 y_1 + x_2 y_2 + 1)^2 =$$

$$= (x_1 y_1)^2 + (x_2 y_2)^2 + 1^2 + 2 x_1 y_1 x_2 y_2 + 2 x_1 y_1 \cdot 1 + 2 x_2 y_2 \cdot 1 =$$

$$= (x_1 y_1)^2 + (x_2 y_2)^2 + 1 + 2 x_1 y_1 x_2 y_2 + 2 x_1 y_1 + 2 x_2 y_2 =$$

$$= (x_1 y_1)^2 + (x_2 y_2)^2 + 1 + (\sqrt{2} x_1 x_2)(\sqrt{2} y_1 y_2) + \underbrace{2 x_1 y_1}_{(\sqrt{2}x_1)(\sqrt{2}y_1)} + \underbrace{2 x_2 y_2}_{(\sqrt{2}x_2)(\sqrt{2}y_2)} =$$

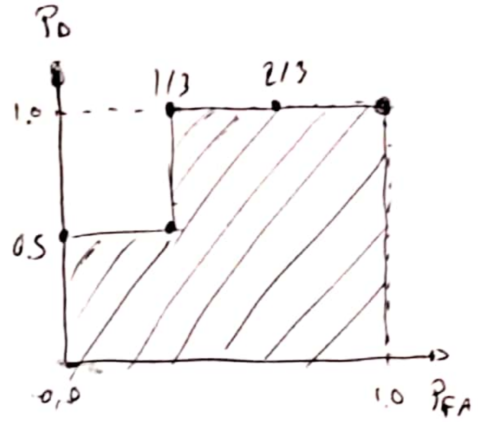
$$= (x_1^2, x_2^2, 1, \sqrt{2} x_1 x_2, \sqrt{2} x_1, \sqrt{2} x_2) \cdot (y_1^2, y_2^2, 1, \sqrt{2} y_1 y_2, \sqrt{2} y_1, \sqrt{2} y_2)$$

$$\begin{pmatrix} u \\ v \end{pmatrix} \to \begin{pmatrix} 1 \\ u^2 \\ v^2 \\ \sqrt{2}uv \\ \sqrt{2}u \\ \sqrt{2}v \end{pmatrix}$$

5. a)

Sample	Prediction	True Label
1	0.8	1
2	0.5	1
3	0.6	0
4	0.4	0
5	0.2	0

		Actual: Detection	Actual: No detection
Prediction	Detection	T +	F +
	No detection	F −	T −



(1) ~~all~~ $T < 0.2$: $P_D = 1.0$ $P_{FA} = 1.0$

2	3
0	0

(2) $0.2 \le T < 0.4$: $P_D = 1.0$ $P_{FA} = \frac{2}{3} = 0.67$

2	2
0	1

(3) $0.4 \le T < 0.5$: $P_D = 1.0$ $P_{FA} = \frac{1}{3} = 0.33$

2	1
0	2

(4) $0.5 \le T < 0.6$: $P_D = 0.5$ $P_{FA} = 0.33$

1	1
1	2

(5) $0.6 \le T < 0.8$: $P_D = 0.5$ $P_{FA} = 0.0$

1	0
1	3

(6) $T > 0.8$: $P_D = 0.0$ $P_{FA} = 0.0$

0	0
2	3

$$AUC = \frac{1}{3}\cdot\frac{1}{2} + \frac{2}{3}\cdot 1 = \frac{1}{6} + \frac{2}{3}\frac{2}{2} =$$

~~= 0.16 + 0.67 = 0.83~~

$$= \frac{5}{6} = 0.8\overline{3}$$

Scanned by CamScanner