MACHINE LEARNING Course 2

Advanced Learning Algorithms.

12/2/24

Speech recognition (Neural networks) → Improve performance of data.

neuron → layer (Hidden layers)
↳ (activation)

ANN ~~is a~~ mathem. model of biological neuron. $\vec{x} \to [\,\circ\ \circ\ \circ\,] \to [\,\circ\,] \to \hat{y}$
① ②

$\vec{a}^{[2]} \to \boxed{w_2^{[3]}, b_2^{[3]}} \to a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot \vec{a}^{[2]} + b_2^{[3]})$

$$\boxed{a_j^{[l]} = g(\vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]})}$$

activation value of layer l, unit (neuron) j

sigmoid fn → activation function. (ANN)

- forward Propagation : layers keep on decreasing in activation fns.

Tensor flow → ML package, Keras → integrated TF for layer centric interface

$g(\vec{w}\vec{x}+b)$ w → weight b → bias

through numpy, we created single layer, T.F. allows multi-layer, model.

layer_1 = Dense(unit=3, activation='sigmoid')

a1 = layer_1 (~~x~~)

tf.Tensor (— —), shape (1,3)

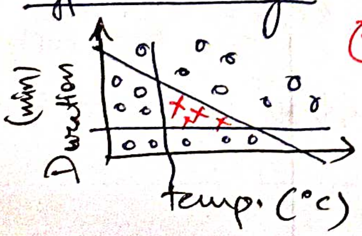
a1. numpy ().

13/2/24

model = Sequential ([layer_1, layer_2])

plain stack of layers — i/p one tensor ↓ one o/p tensor

Coffee Roasting



Ⓧ Good roast probability

undercooked
too short duration
overcooked.

Lab 2

Matrix Multiplication.

$\vec{a} \cdot \vec{w}$ ~~or~~ $\vec{a}^T \times \vec{W}$

np.matmul (A_T, W) + B → numpy function

$\boxed{Z = a_T \times W}$ ← this is not dot-product

np. matmul (a_T, W)

- Numpy Broadcasting

↳ a: 4×1, b: 1 → c = a+b = 4×1

* np. array ( ). reshape(−1, 1) → row matrix

np. array( ). reshape(1, −1) → column matrix

Week 2

C2 W2

Model Training Steps

① how to compute output from given inputs & parameters.

② specify loss & cost.

③ Train on data to minimise J(w, b)

In Neural Network.

(1) model = Sequential (Dense L1, Dense L2)

(2) compile. loss (Binary CrossEntropy())

(3) model. fit

ReLU → an activation function, → Rectified Linear.
$g(z) = \max(0, z)$.

Linear Activation fn → $g(z) = z$
$z = wx + b$

sigmoid → $\frac{1}{1+e^{-z}} = g(z)$

Relu is most commonly used. It is fast.

- Gradient descent is slow when g(z) is flat

Use ReLU for Hidden layer.

Sigmoid → 0,1

Linear → Regression (+/−)

ReLU ↳ y = 0/+

120g-carb
1.5 kg/Body weight protein
1.5

60 - 70g.
60 - 1.5 - 90g. pool
90 g for carbo

Why to use activation function in neural network

Don't use linear activations in hidden layer. ⊛

Linear activation is just like ~~logistic~~ No Activat fn. {like linear regression}

Do read ReLU Lab. week 2

ReLU has non linear behaviour for $x \le 0$

⊛ **Multi Class Classification**

target y can take on more than two categories (0 to 9)

Variety of diseases

SOFTMAX regression algorithm.

generalisation of logistic regression in multi class context:

$P(y=i \mid \vec{x})$ i is one of the outputs possible

$$a_i = \frac{e^{z_i}}{e^{z_i} + e^{z_j} + e^{z_k} + \dots}$$

$$\boxed{\sum_{i=1}^{n} a_i = 1}$$ similar to Logistic reg.

for $n=2$, it is same as Logistic Reg.

a_1 ✓ $\boxed{a_2 = 1 - a_1}$

Softmax Regression → $\boxed{\text{loss} = -\log a_i}$ if $(y=i)$

$\boxed{y = -\log a}$

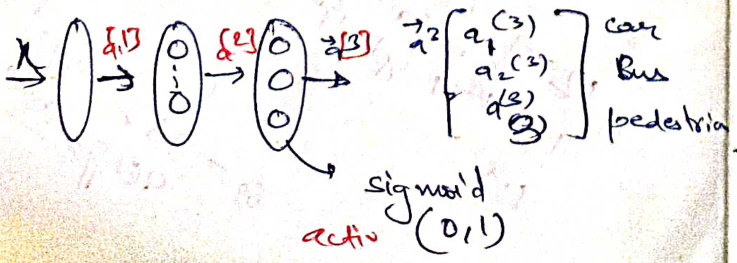
⊡ Learnt mnist network using softmax. (0 to 9)

Make code more refined.

Cross Entropy Loss → $\boxed{-\log(a_i)}$

Multi Label Classification

car, bus pedestrian



SoftMax function

$$a_j = \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_k}}$$

only one of the loss (a_j) contribute to actual loss.

epoch → training of neural network with all training data for 1 cycle

---

Gradient Descent

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j}(J(w,b))$$

↓ learning Rate

Adam algorithm → go faster → increase α

go slow → dec α

Adaptive Movement Estimation

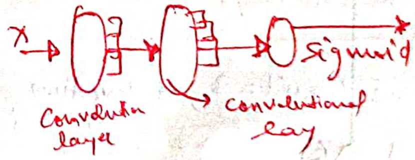
$$\boxed{w_j = w_j - \alpha_j \frac{\partial}{\partial w_j}(J(w,b))}$$

if oscillating it is fast, so dec α

If growing in one direction, speed up.

GD → optimisation algo. used to train model by minimising error b/w predicted & actual ~~Result~~

Convolutional Neural Network.



ECG

Heart disease

look at different groups of inputs.

SoftMax activation in Multiclass Classific

Sympy library in python used for differential calculus operation.

- Chain rule understood via back propagation.

(Optional) $\boxed{\frac{dJ}{dw} = \frac{dJ}{da} \cdot \frac{da}{dw}}$

{ Week 2 over }

Week 3   C2 W3

Improving the model.

Diagnostic → a test that we run to gain insight into what is / isn't working with algorithm, to gain guidance into improving its performance.

- In a given dataset, train model to 70% & 30% test on model.

↓ (Better)

Training set	Cross Validation	Test Set
(60%)	(20%)	(10%)

find (degree) model with lowest CV error.

This is called Model Selection.

Bias & Variance

high bias → underfit

high variance → overfit.

$J_{train} < J_{cv}$

high bias → doing bad at training set

high var. → doing worse at CV set.

- Bias → training set error
- Variance → test set error.

low bias	high variance	overfit
high bias	high variance	underfit
low bias	low variance	Balanced

Cross Validation →

technique to evaluate performance of a model on unseen data

- Regression → predict cts. values
- Classification → classify data

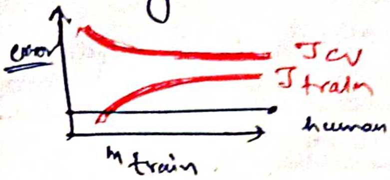
dependent
independent
variables

Feature Scaling

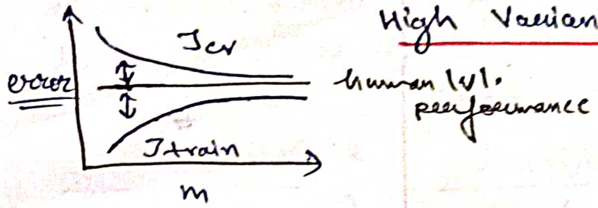
transform values of features of variables in dataset to a similar scale.

Regularization → method to reduce overfitting.

for high bias, getting more training data will not help much



Learning Curves



High Variance.

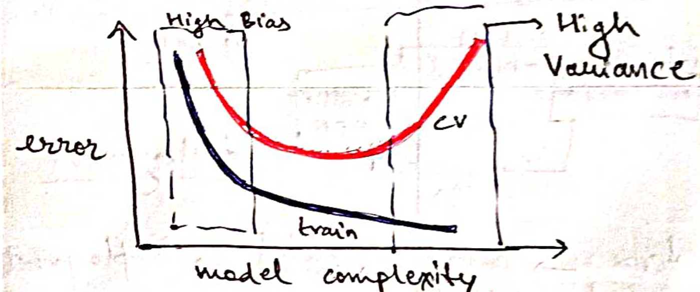
adding more examples will help

λ → regularization parameter.

inc λ → high variance fixed (make underfit)

dec λ → fixes high bias (make overfit)

- pick lowest CV error.

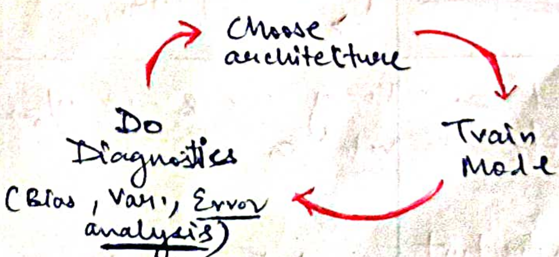


- Large Neural networks are low bias machines

regularize larger NN optimally.

- Most important → variation of bias and variance.

See test of Bias and Variance.



Email Routing → ensures that right message goes to the right recipient.

Data Augmentation

modifying an existing training example to create a new training example.

→ use distortions

Artificial ~~Data~~ Synthesis for photo OCR

Transfer Learning

knowledge learned from a task is reused in order to boost performance

- Supervised Pre-training

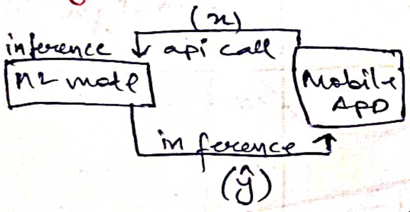
help to learn generic basic structures / data.

- Fine Tuning

↳ take a pre-trained model and train on smaller target set (our data)

Sharing of code / model

Cycle of ML project



ML ops — operation.

deployment

Scope of project → Data collection → Model Training → Deployment

* Monitor & Maintainence

* Data Augmentation

Option 1 → only train output layers param

Option 2 → train all parameters.

ethics of ML → use cautiously, deepfake

Skewed Datasets → ratio of +ve & -ve o/ps is not 50-50.

↓

very important

→ F_1 score

→ precision and recall

Actual

		1	0
Pred	1	TP	FP
	0	FN	TN

classification threshold

Week 4

- Numpy → scientific computing
- Scikit Learn → Data Mining
- Tensorflow → ML platform.

Standard Scaler → mean and standard deviation

^ transforms it.

More data improves generalization.

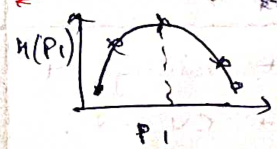
↓

ability to adapt properly to new, prev. unseen data drawn from same dist^n as one used to create the model

Decision Trees

A non-parametric supervised learning algorithm, used for classification and regression tasks.

Entropy → measure of purity.



$H(P_1) = -p_1 \log_2 P_1 - p_0 \log_2 P_0$

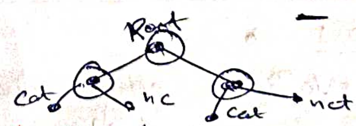
where $\boxed{P_0 = 1 - P_1}$

Information Gain

method of reduction of entropy.

→ increase purity of subset of data.

$H(P_1^{root}) - (w_{left} H(P_1\,left) - w_{right} H(P_1\,right))$



→ $P_1\,left = x/a$ $P_1\,right = y/b$

→ $w_{left} = \frac{a}{a+b}$ $w_{right} = \frac{b}{a+b}$

where $h_{root} = \frac{x+y}{a+b}$

one hot encoding

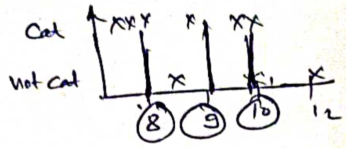
lower entropy, higher purity

Week 4 C2 W4

↳ continuous valued features

Select the threshold that gives you highest information gain. using Gradient Descent

eg: weight of animal.



weights

Criteria to stop splitting

- when tree has reached max. depth.
- when no. of examples in a node is below a threshold.

Regression Tree → feature prediction from another feature.

↓

reduction in variance is used, similar to information gain.

Choose feature with largest red^n in variance

PRECISION → accuracy of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall → It calculates how many of actual positive cases were correctly predicted by model.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Balance b/w Precision & Recall →

F_1 score = Harmonic Mean (Precision, Recall)

$$\boxed{F_1 \text{ score} = \frac{2}{1/P + 1/R} = \frac{2PR}{P+R}}$$

24-04-2024 Wed

Precision → Predictions mein kitne true +ve

Accuracy of [Recall]

true +ve mein kitne predicted +ve

Tree ensemble Part 2

↳ combines multiple decision trees to make better predictions

Sampling with Replacement

Random Trees → ML algorithm used for classification & regression

contains no. of decision trees on various subsets and takes avg. for better accuracy.

Boosting Algo

↳ convert weak learners [classifiers] to strong ones.

- XG Boost → Xtreme Gradient Boosting. → open source boosted trees.

Decision Trees →

- work well for tabular / structured data
- Fast • Expensive
- unsuitable for images, audio data.

Rest use Neural Network → transfer learning / all types of data / train multiple NN simultan / may be slower.

* Entropy and information gain are key in Decision Trees.

decision at leaf node

⊛ utils.py is a helper function.