FUNDAMENTALS OF DEEP LEARNING

How a Neural Network Trains

Muljono

UNIVERSITAS DIAN NUSWANTORO

TENTANG SAYA

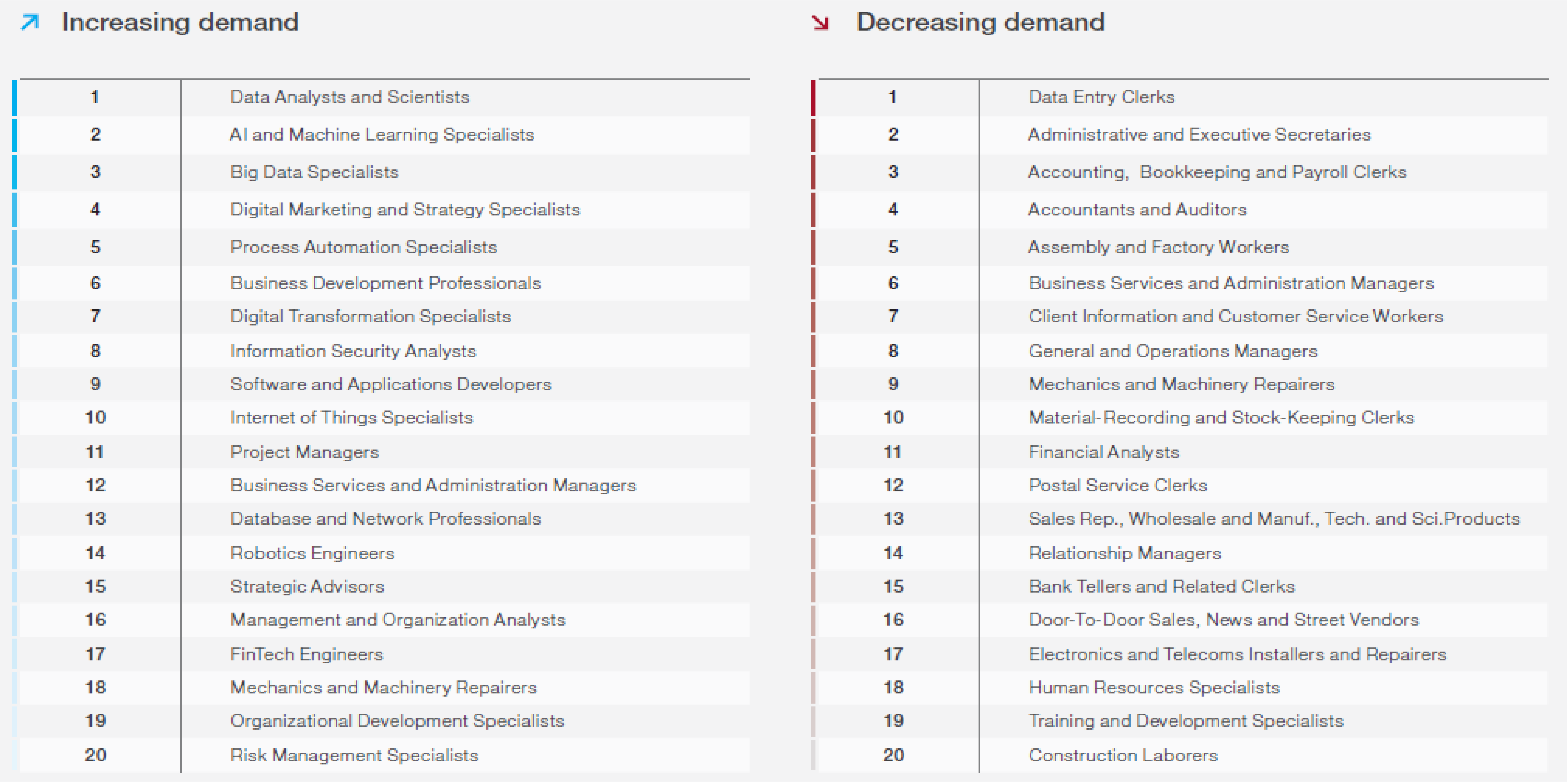
Latar Belakang Pendidikan:

- S1: Matematika Universitas Diponegoro Semarang
- S2: Teknik Informatika STTIBI Jakarta
- S3: Teknik Elektro ITS Surabaya

Riwayat/Pengalaman Pekerjaan:

- Anggota APTI (Asosiasi Profesi Telematika Indonesia)
- Dosen Universitas Dian Nuswantoro Semarang (Associate Professor)
- Post-Doctoral di Universtas Gadjah Mada Yogyakarta
- Sandwich Program di Tokyo University of Technology, Tokyo, Japan
- Teaching Mobility Program Erasmus + di University of Split, Croatia
- Workshop Big Data Analaysis dan IOT, Big Data Alibaba, Fozhou Polythechnic, Fouzhou China

TOP 20 JOB ROLES IN INCREASING AND DECREASING DEMAND



↗ Increasing demand

1	Data Analysts and Scientists
2	AI and Machine Learning Specialists
3	Big Data Specialists
4	Digital Marketing and Strategy Specialists
5	Process Automation Specialists
6	Business Development Professionals
7	Digital Transformation Specialists
8	Information Security Analysts
9	Software and Applications Developers
10	Internet of Things Specialists
11	Project Managers
12	Business Services and Administration Managers
13	Database and Network Professionals
14	Robotics Engineers
15	Strategic Advisors
16	Management and Organization Analysts
17	FinTech Engineers
18	Mechanics and Machinery Repairers
19	Organizational Development Specialists
20	Risk Management Specialists

↘ Decreasing demand

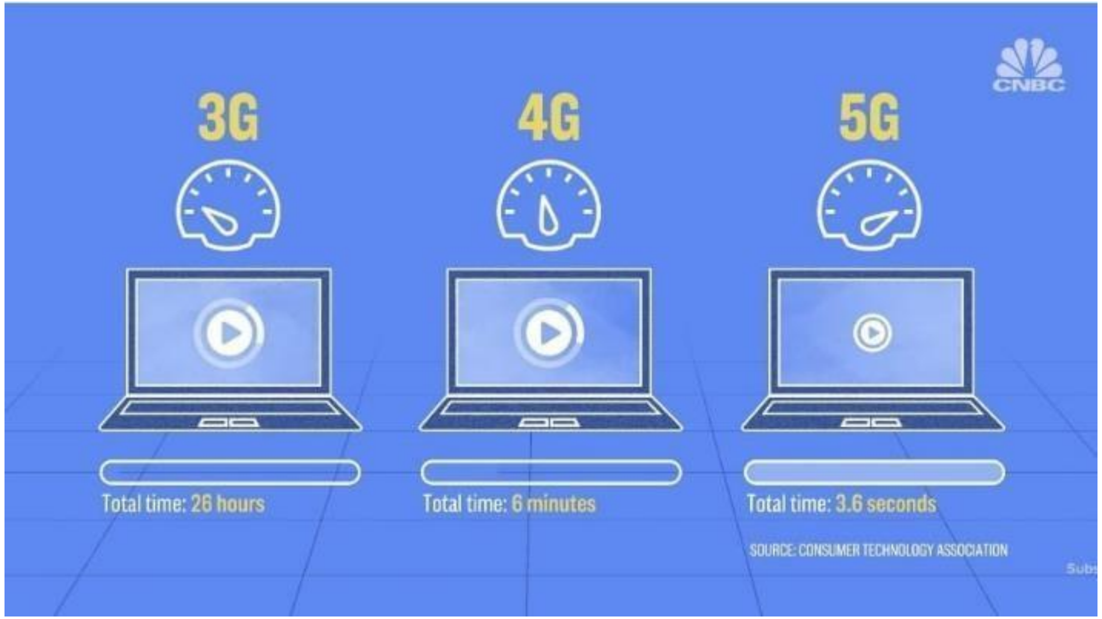
1	Data Entry Clerks
2	Administrative and Executive Secretaries
3	Accounting, Bookkeeping and Payroll Clerks
4	Accountants and Auditors
5	Assembly and Factory Workers
6	Business Services and Administration Managers
7	Client Information and Customer Service Workers
8	General and Operations Managers
9	Mechanics and Machinery Repairers
10	Material-Recording and Stock-Keeping Clerks
11	Financial Analysts
12	Postal Service Clerks
13	Sales Rep., Wholesale and Manuf., Tech. and Sci.Products
14	Relationship Managers
15	Bank Tellers and Related Clerks
16	Door-To-Door Sales, News and Street Vendors
17	Electronics and Telecoms Installers and Repairers
18	Human Resources Specialists
19	Training and Development Specialists
20	Construction Laborers

Source : Future of Jobs Survey 2020, World Economic Forum

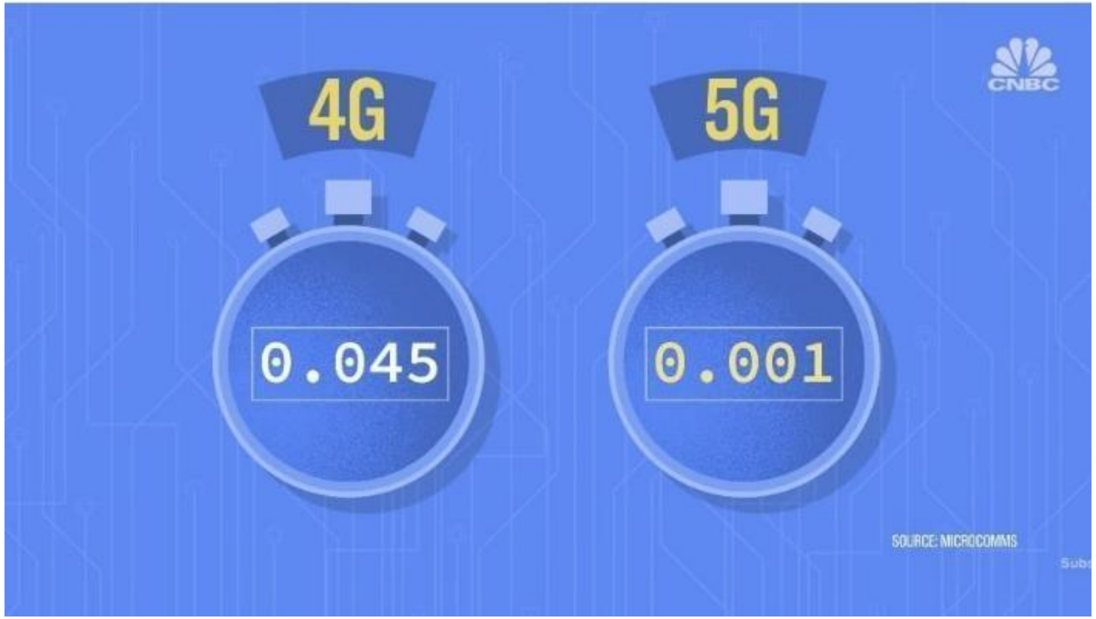
TECHNOLOGY TREND:

- **Technology 5G**
- **Cyber Security**
- **AI (Artificial Intelligence)**

THE RISE OF TECHNOLOGY 5G



SOURCE: CONSUMER TECHNOLOGY ASSOCIATION



SOURCE: MICROCOMMS

THE RISE OF CYBER SECURITY

THE GREAT ARTIFICIAL INTELLIGENCE AWAKENING

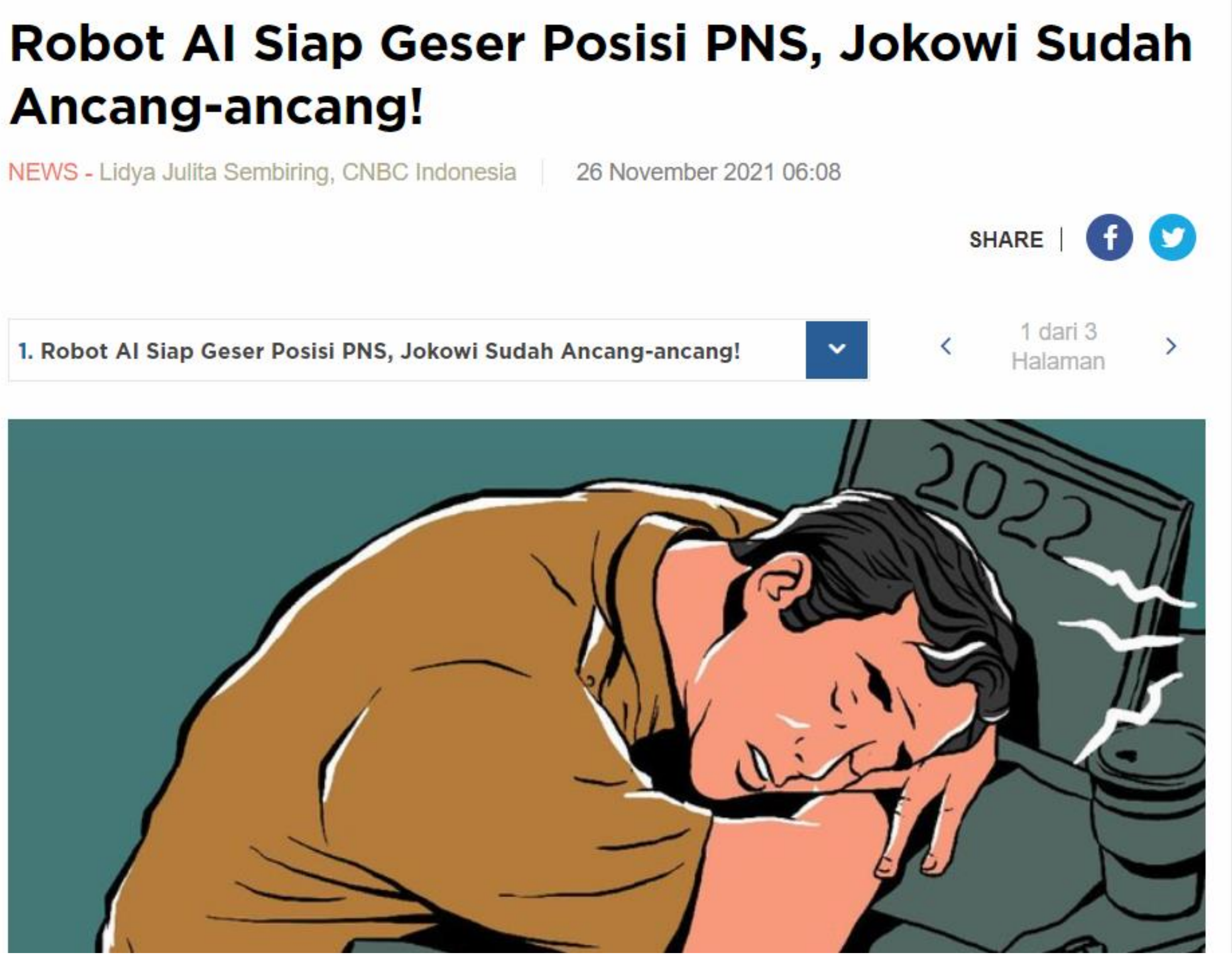
"AI is not the next big thing — it’s the current big thing !

Welcome to the Artificial Intelligence era"

"KITA KEJAR-KEJARAN, SIAPA KUASAI **AI** BERPOTENSI KUASAI DUNIA"

Joko Widodo

President of Indonesia

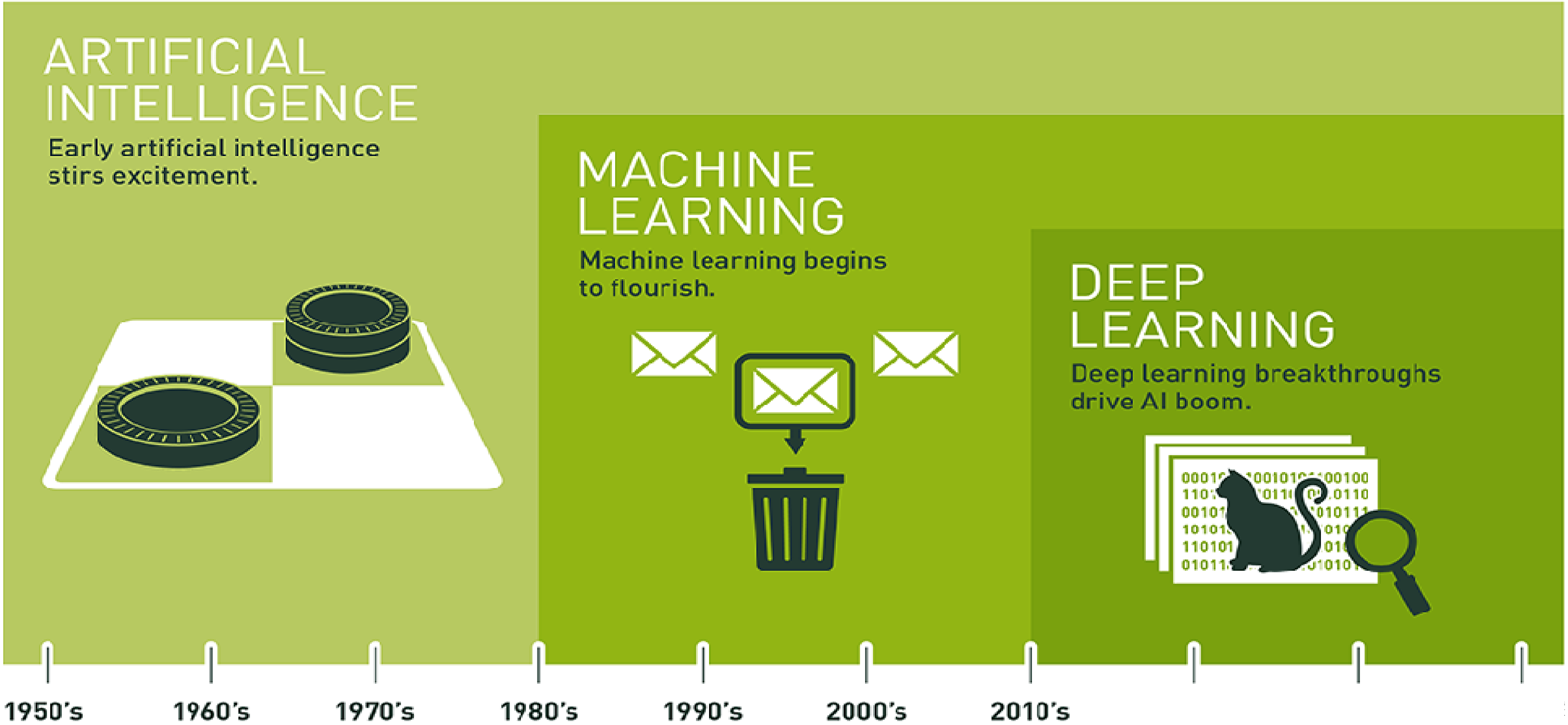


Robot AI Siap Geser Posisi PNS, Jokowi Sudah Ancang-ancang!

NEWS - Lidya Julita Sembiring, CNBC Indonesia | 26 November 2021 06:08

1. Robot AI Siap Geser Posisi PNS, Jokowi Sudah Ancang-ancang!

RELATIONSHIP BETWEEN AI, ML, DL



ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.

MACHINE LEARNING

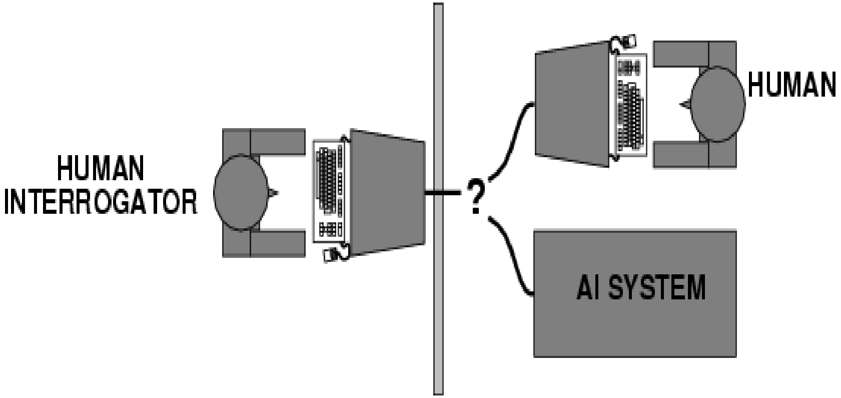
Machine learning begins to flourish.

DEEP LEARNING

Deep learning breakthroughs drive AI boom.

ARTIFICIAL INTELLIGENCE

Machines that think rationally.	Machines that act like humans
Machines that think like humans.	Machines that act rationally



Turing (1950) "Computing machinery and intelligence”

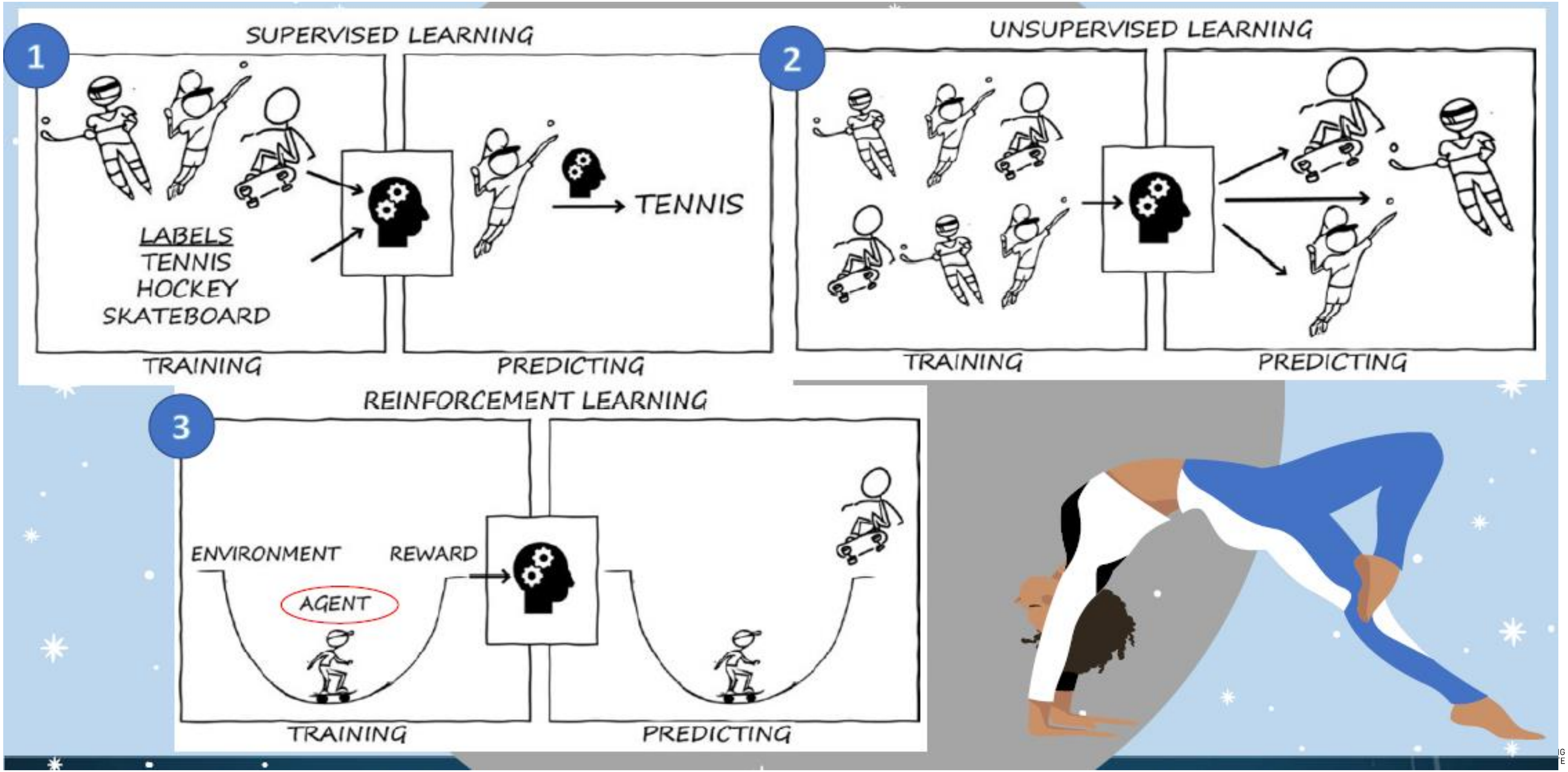
MACHINE LEARNING

- Ability of computers to “learn” from “data” or “past experience”

- data: Comes from various sources such as sensors, domain knowledge, experimental runs, etc.

- learn: Make *intelligent* predictions or decisions based on data by optimizing a model

AREA MACHINE LEARNING



PRE-REQUISITE TO MACHINE LEARNING

- Probability
  - distribution, random variable, expectation, conditional probability, variance, density
- Linear algebra
  - matrix multiplication
  - eigenvector
- Basic programming (in Python)

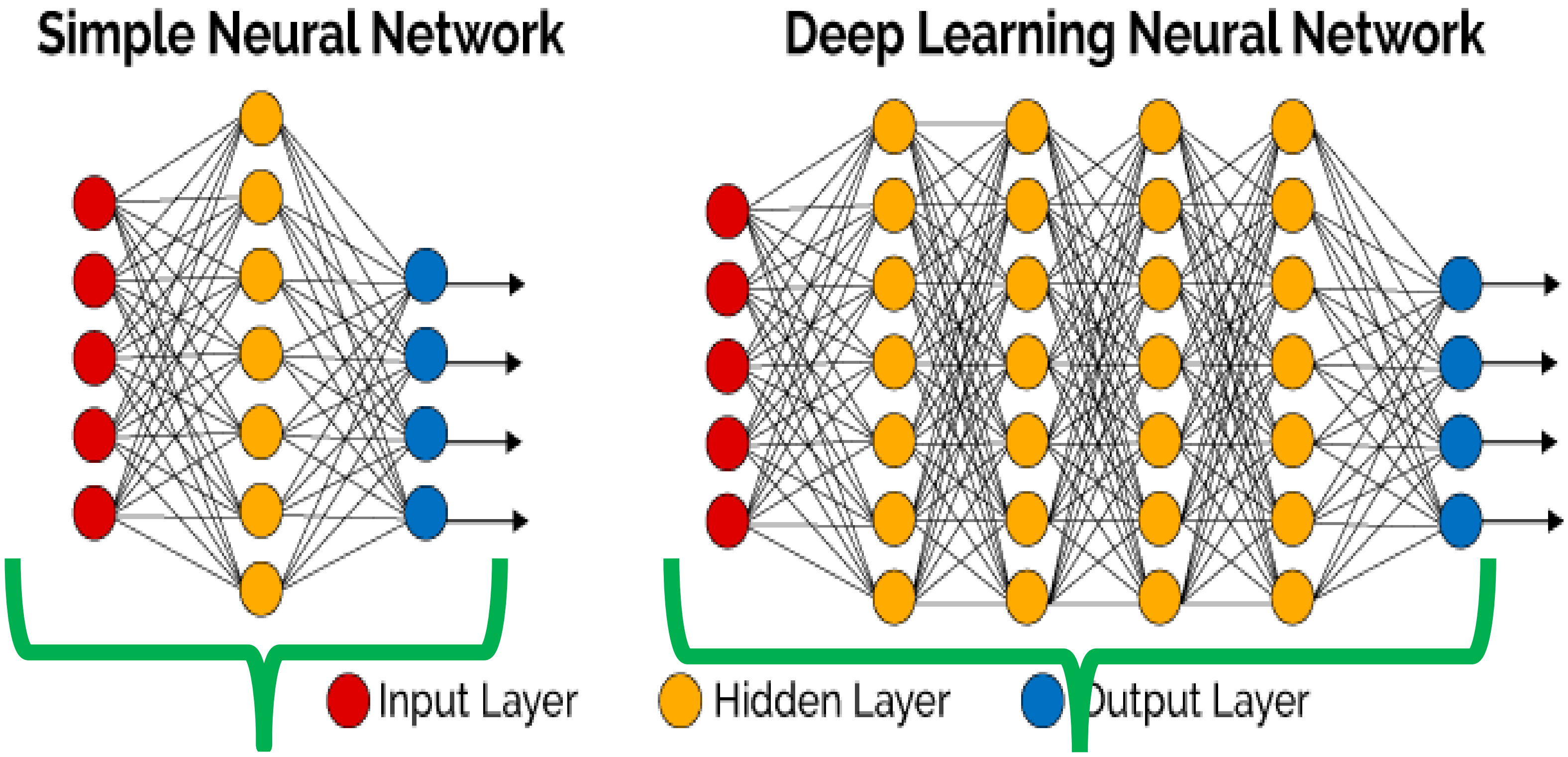
😉

DEEP LEARNING

Deep learning is a subset of **machine learning** where artificial **neural networks**, algorithms inspired by the human brain, **learn** from large amounts of data.

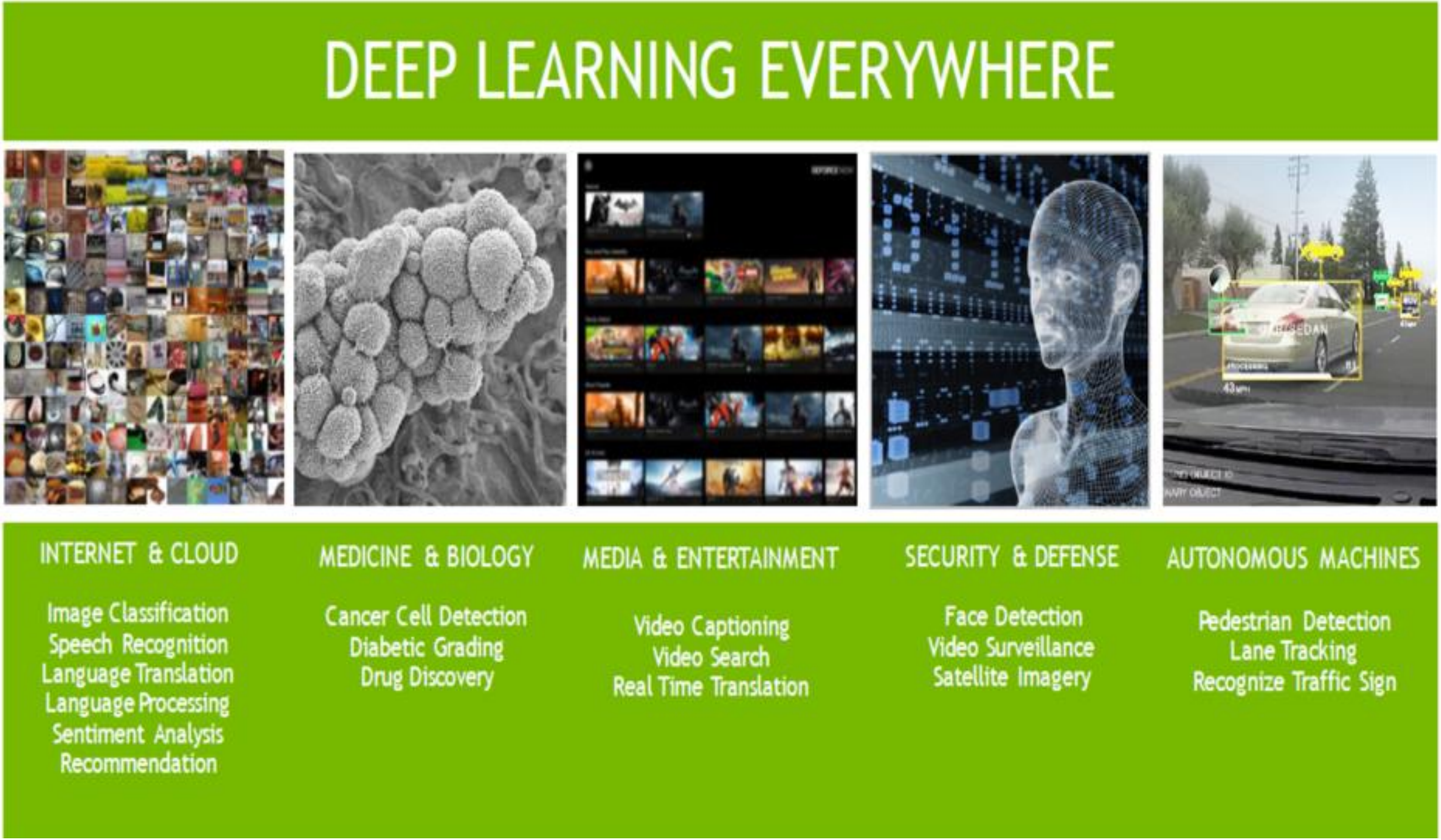
Deep learning allows machines to solve complex problems even when using a data set that is very diverse, unstructured and inter-connected.

“DEEP” LEARNING ?



1 hidden layer

many hidden layers



DEEP LEARNING EVERYWHERE

INTERNET & CLOUD

Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation

MEDICINE & BIOLOGY

Cancer Cell Detection
Diabetic Grading
Drug Discovery

MEDIA & ENTERTAINMENT

Video Captioning
Video Search
Real Time Translation

SECURITY & DEFENSE

Face Detection
Video Surveillance
Satellite Imagery

AUTONOMOUS MACHINES

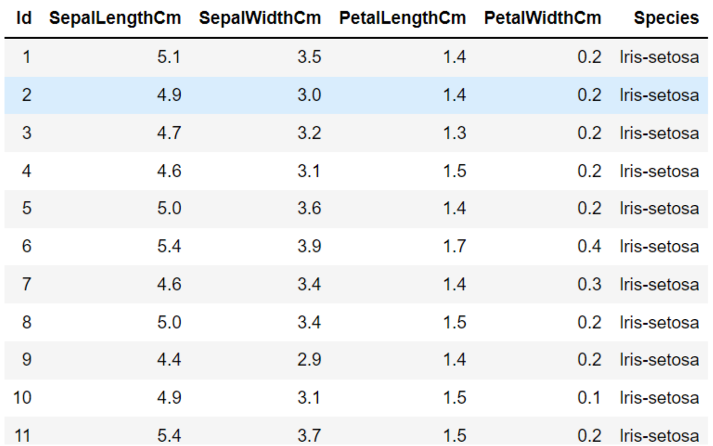
Pedestrian Detection
Lane Tracking
Recognize Traffic Sign

Source : developer.nvidia.com/deep-leaning-course

UNSTRUCTURED DATA VS STRUCTURED DATA

a speech signal
20000
10000
0
−10000
−20000
0
40000
50000
60000
Fearful
Angry
Sad
Happy
Disgusted
Surprised

unstructured data



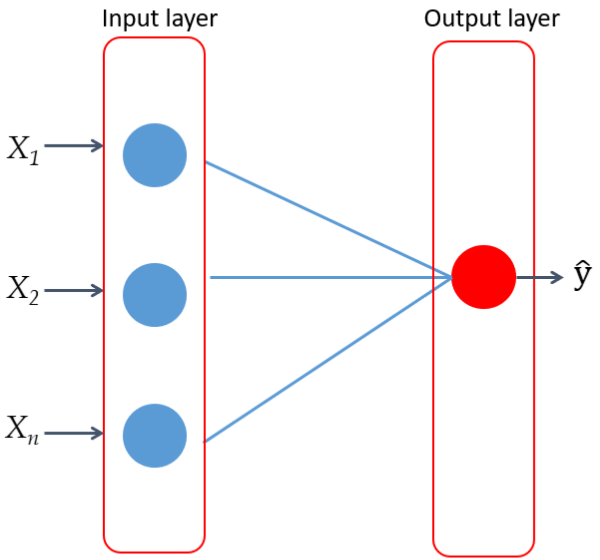
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa

structured data

FUNDAMENTALS OF DEEP LEARNING

Artificial Neural Network

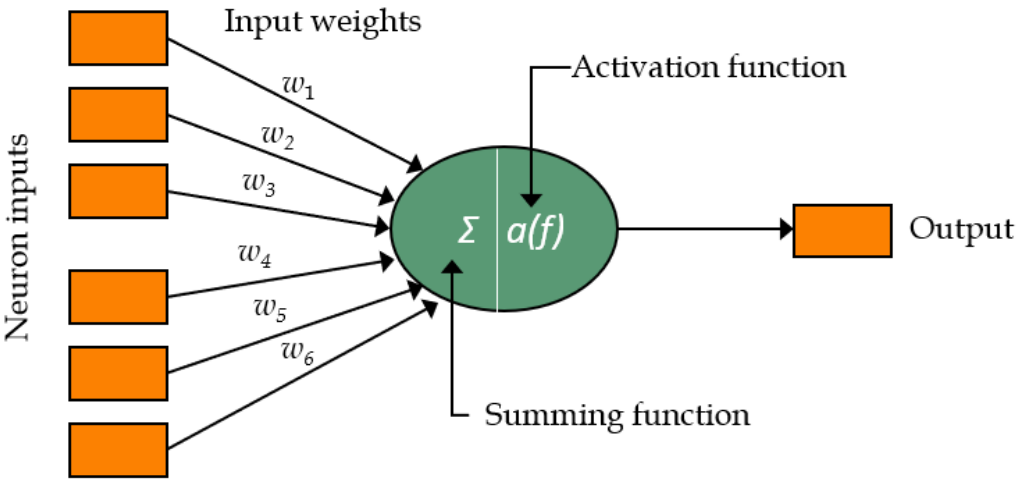
Arsitektur Single-layer Perceptron



Arsitektur single-layer ANN hanya terdiri dari input layer dan output layer

Unit pemrosesan informasi pada ANN sebagai berikut:

- Satu set link berupa neuron dan bobot w
- Fungsi penambah (penggabung linear) untuk mengitung jumlah perkalian bobot terhadap input X
- Fungsi aktivasi $a(\cdot)$



$$f = \sum_{i=1}^{m} w_i x_i + b$$

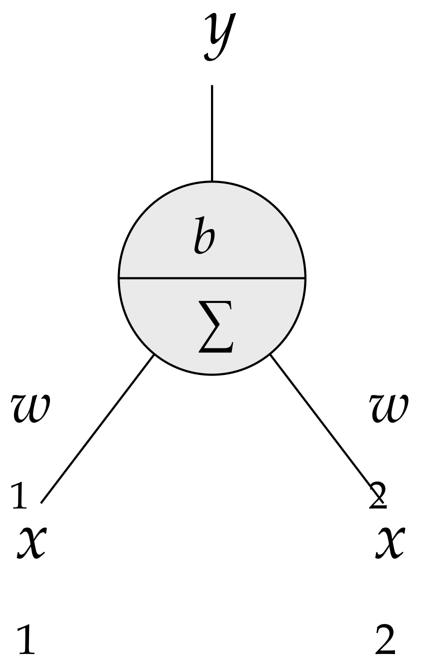
$$y = a(f)$$

Source : Microcredential Associate Data Science – Dikti 2021

Apa yang bisa dilakukan sebuah Neuron ?

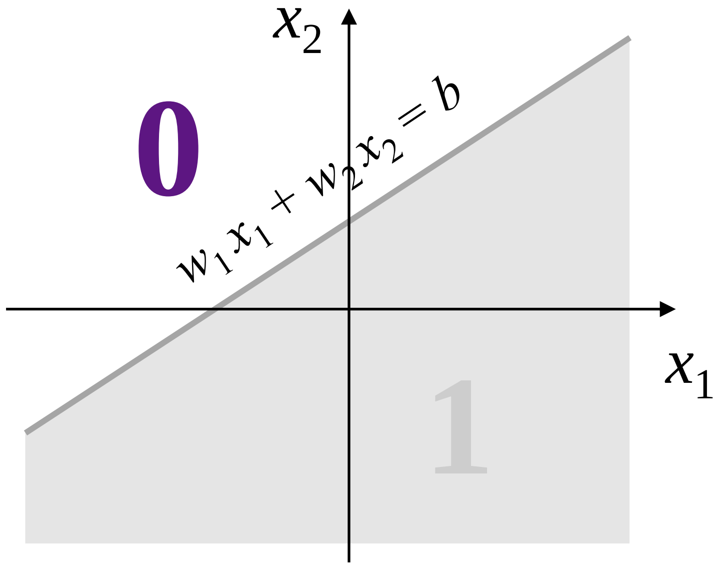
Sebuah neuron pada ANN dapat menyelesaikan permasalahan klasifikasi biner

- Sebagai fungsi pemisah (*hyperspace separation*)
- Sebagai *binary threshold*



$$f(x) = w_1x_1 + w_2x_2 - b$$

$$y = \begin{cases} 1 & f(x) \geq 0 \\ 0 & otherwise \end{cases}$$

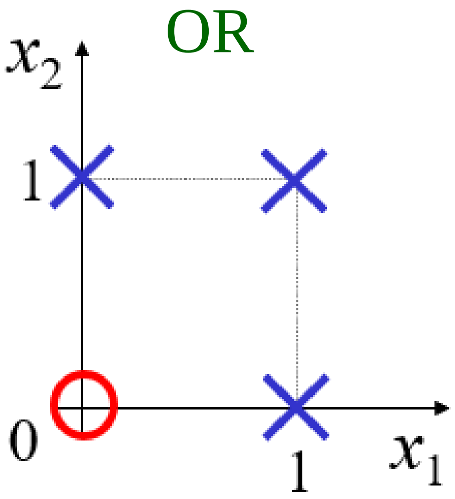


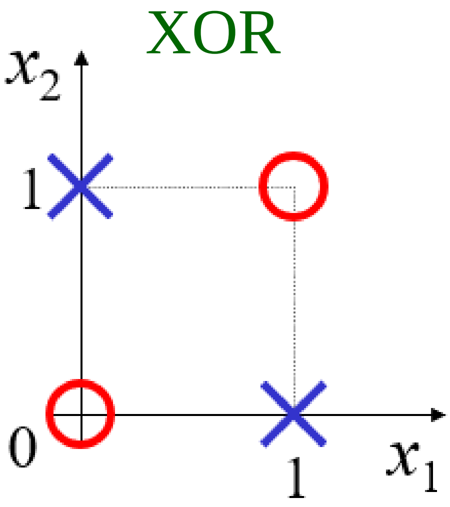
Source : Microcredential Associate Data Science – Dikti 2021

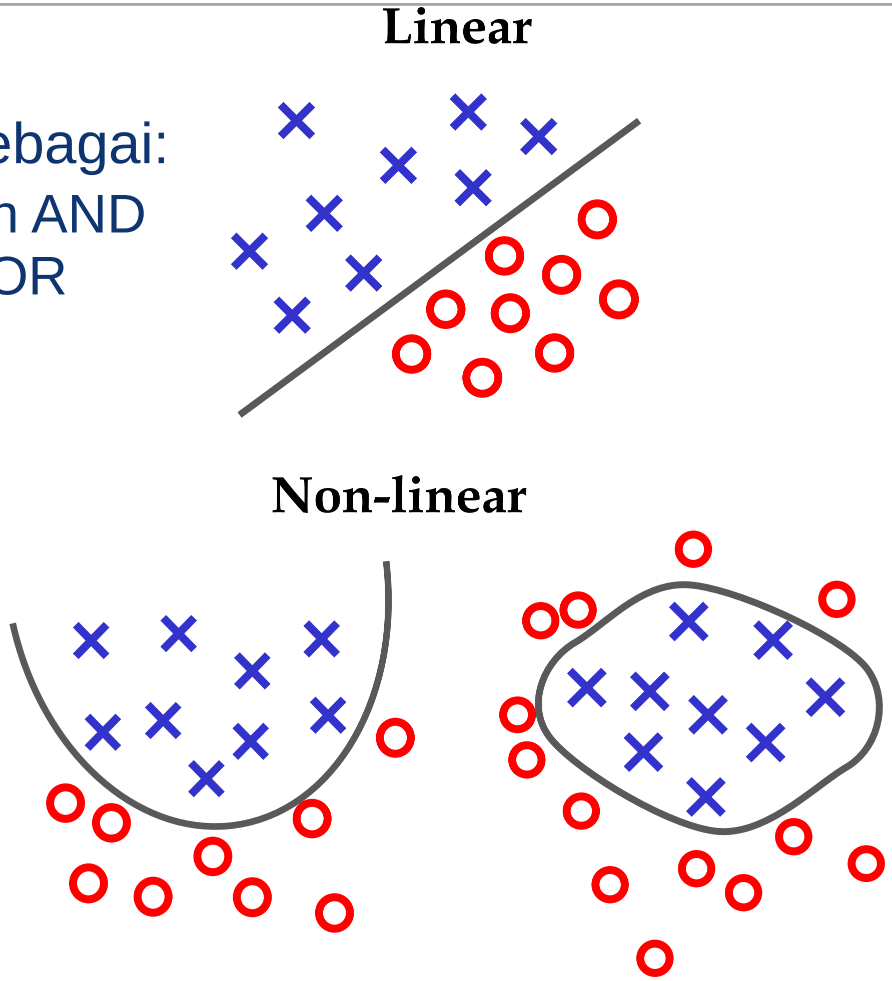
Permasalahan Linear dan Non-Linear

Permasalahan klasifikasi dapat dikategorikan sebagai:

- Permasalahan Linear, misalnya fungsi OR dan AND
- Permasalahan Non-Linear, misalnya fungsi XOR





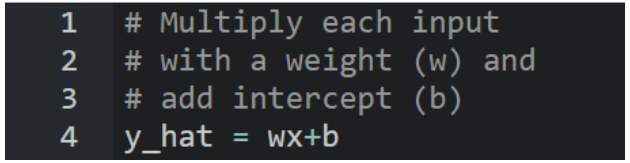


Source : Microcredential Associate Data Science – Dikti 2021

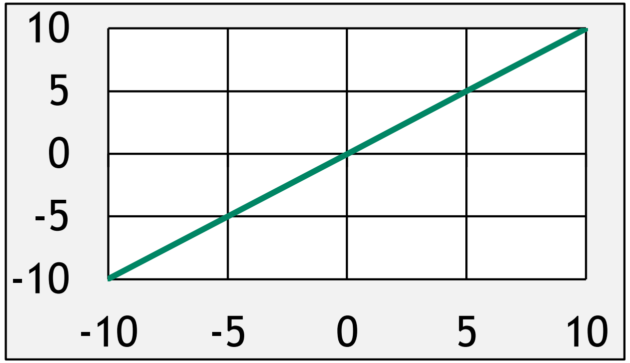
ACTIVATION FUNCTIONS

Linear

$$\hat{y} = wx + b$$

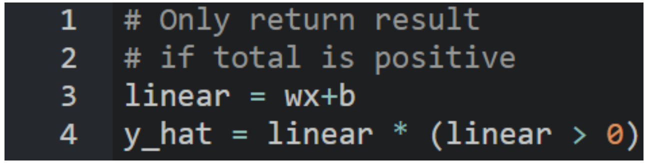


```python
# Multiply each input
# with a weight (w) and
# add intercept (b)
y_hat = wx+b
```

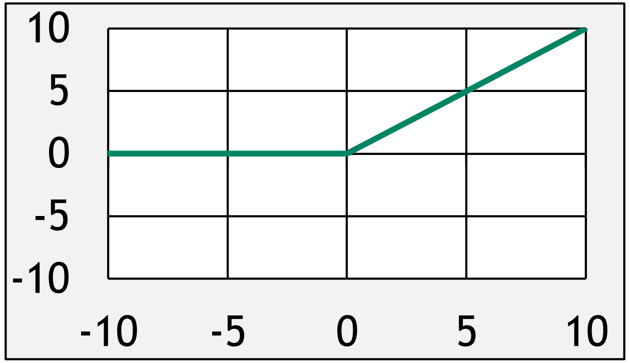


ReLU

$$\hat{y} = \begin{cases} wx + b \; if \; wx + b > 0 \\ 0 \;\; otherwise \end{cases}$$

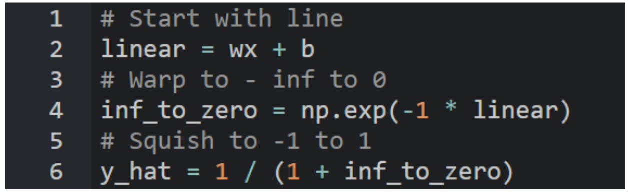


```python
# Only return result
# if total is positive
linear = wx+b
y_hat = linear * (linear > 0)
```

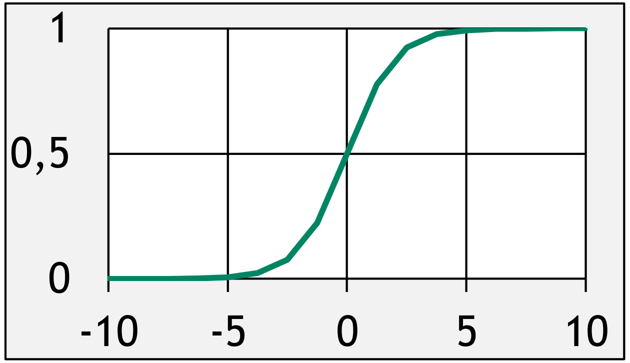


Sigmoid

$$\hat{y} = \frac{1}{1 + e^{-(wx+b)}}$$

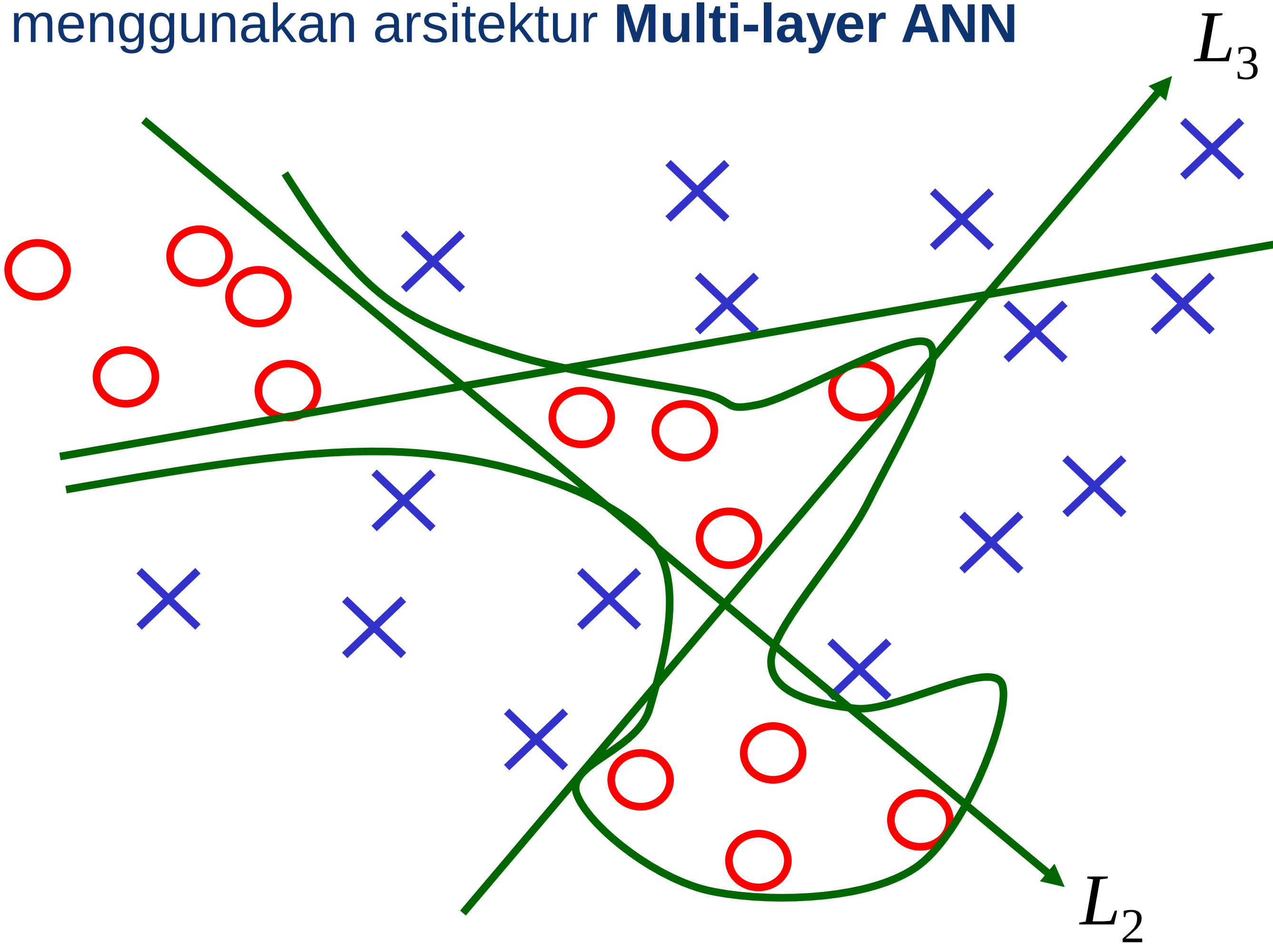


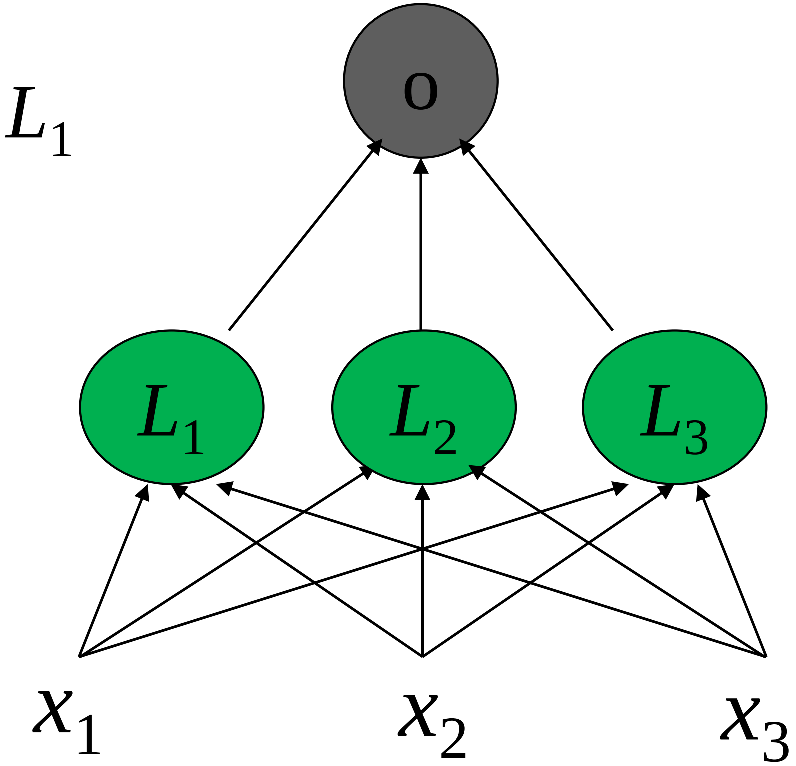
```python
# Start with line
linear = wx + b
# Warp to - inf to 0
inf_to_zero = np.exp(-1 * linear)
# Squish to -1 to 1
y_hat = 1 / (1 + inf_to_zero)
```



Source : developer.nvidia.com/deep-leaning-course

Pada permasalahan non-linear dan permasalahan yang lebih kompleks, menggunakan arsitektur **Multi-layer ANN**

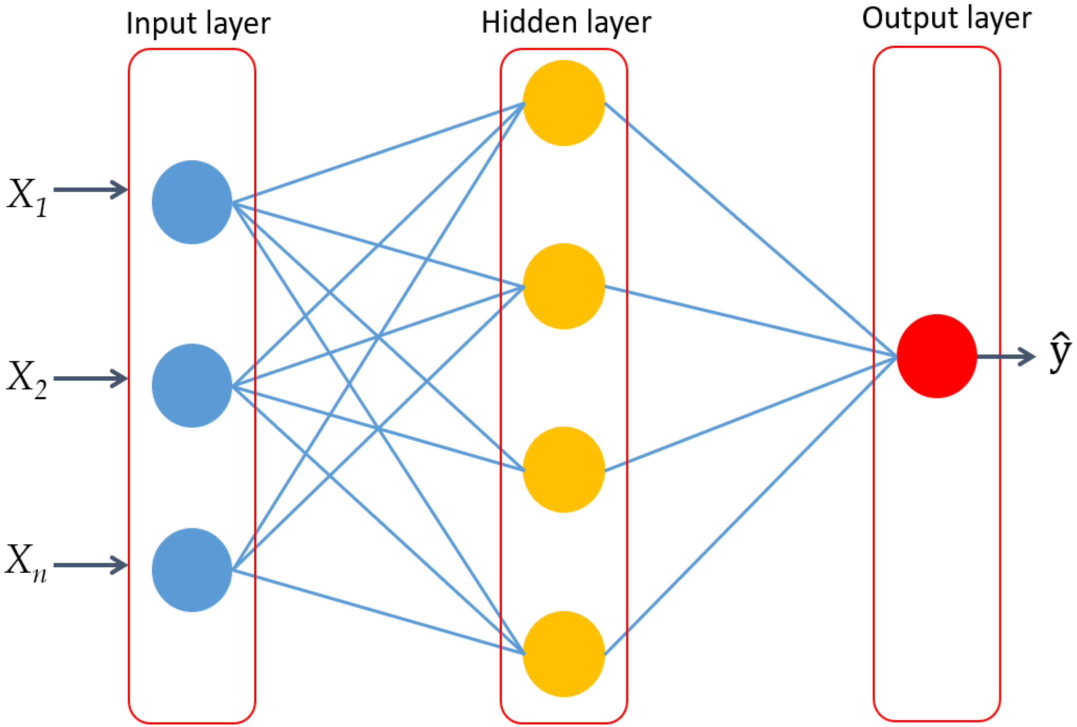




Source : Microcredential Associate Data Science – Dikti 2021

Arsitektur Multi-layer ANN

- Terdiri dari tiga layer yaitu:
  - input layer
  - hidden layer
  - output layer
- Hubungan antar neuron pada ANN merupakan ***fully connected network* (FCN)**
- **Jumlah *hidden layer*** sebaiknya disesuaikan dengan kompleksitas permasalahan
- **Jumlah neuron pada *hidden layer*** umumnya lebih banyak daripada jumlah *neuron* di *output layer*



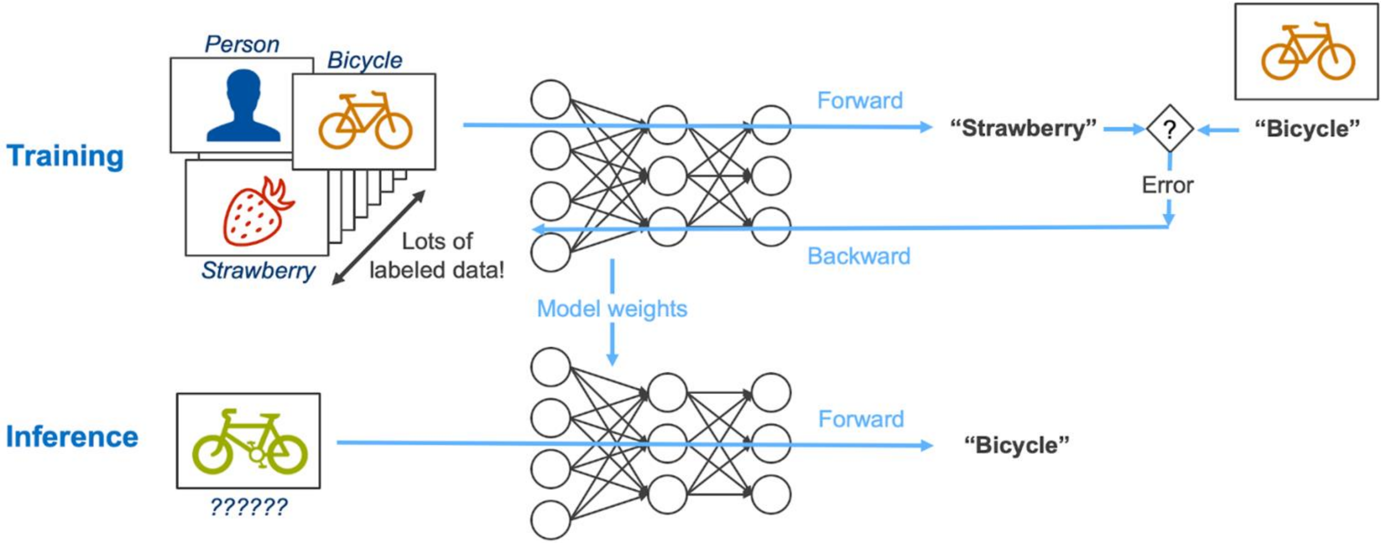
Source : Microcredential Associate Data Science – Dikti 2021

How a Neural Network Trains

Mekanisme Pembelajaran (*Learning*)

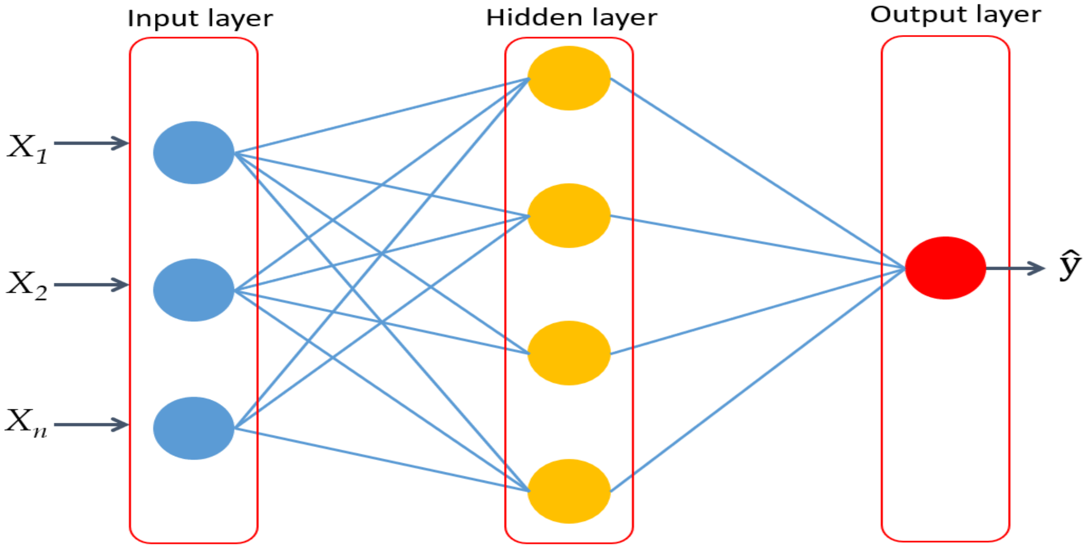
1. Training: learning pada saat pembentukan model.

2. Inferensi: learning saat menggunakan model NN.



Source : Microcredential Associate Data Science – Dikti 2021

Tahapan Pembelajaran Multi-layer Perceptron ANN

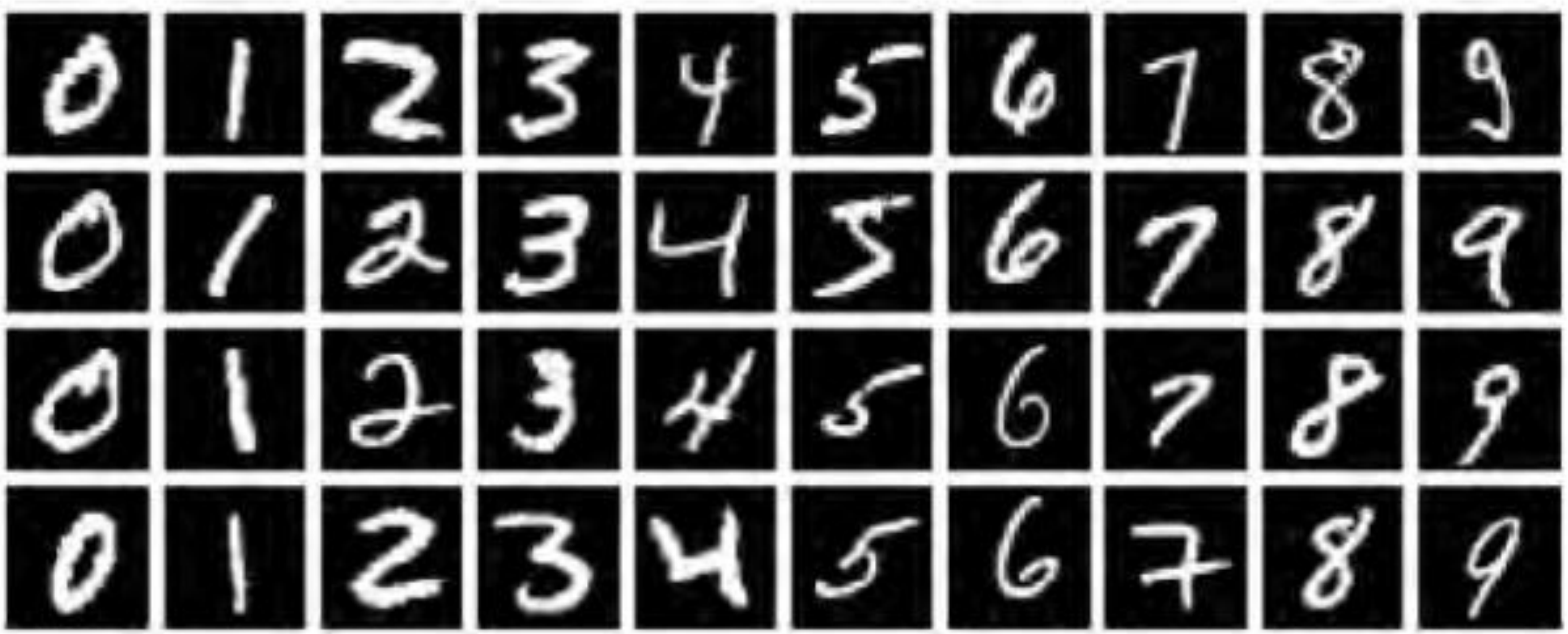


Langkah 0 – Inisialisasi bobot, learning rate, maksimum iterasi
Langkah 1 – Membaca vektor input *X*
Langkah 2 – Lakukan iterasi (*epoch*)
Langkah 3 – Hitung luaran neuron di hidden layer dan output layer
Langkah 4 – Hitung *back propagate error* (pada output layer dan hidden layer)
Langkah 5 – Perbarui semua bobot (pada output layer dan hidden layer)
Langkah 6 – Ulangi langkah 3 – 5 hingga bobot konvergen atau maksimum iterasi
Langkah 7 – Luaran berupa matrik bobot (pada output layer dan hidden layer)

https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/

THE MNIST DATASET

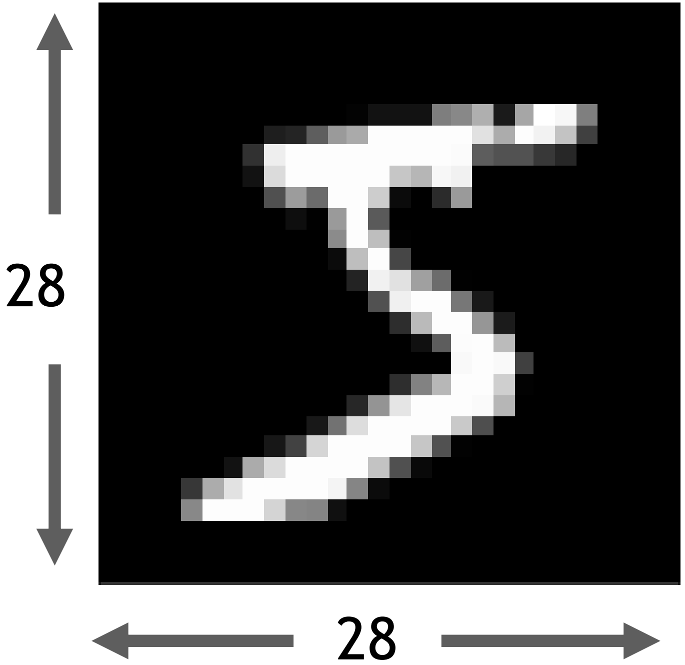
Here are 40 of the images included in the MNIST dataset:



Source : developer.nvidia.com/deep-leaning-course

DATA PREPARATION

Input as an array

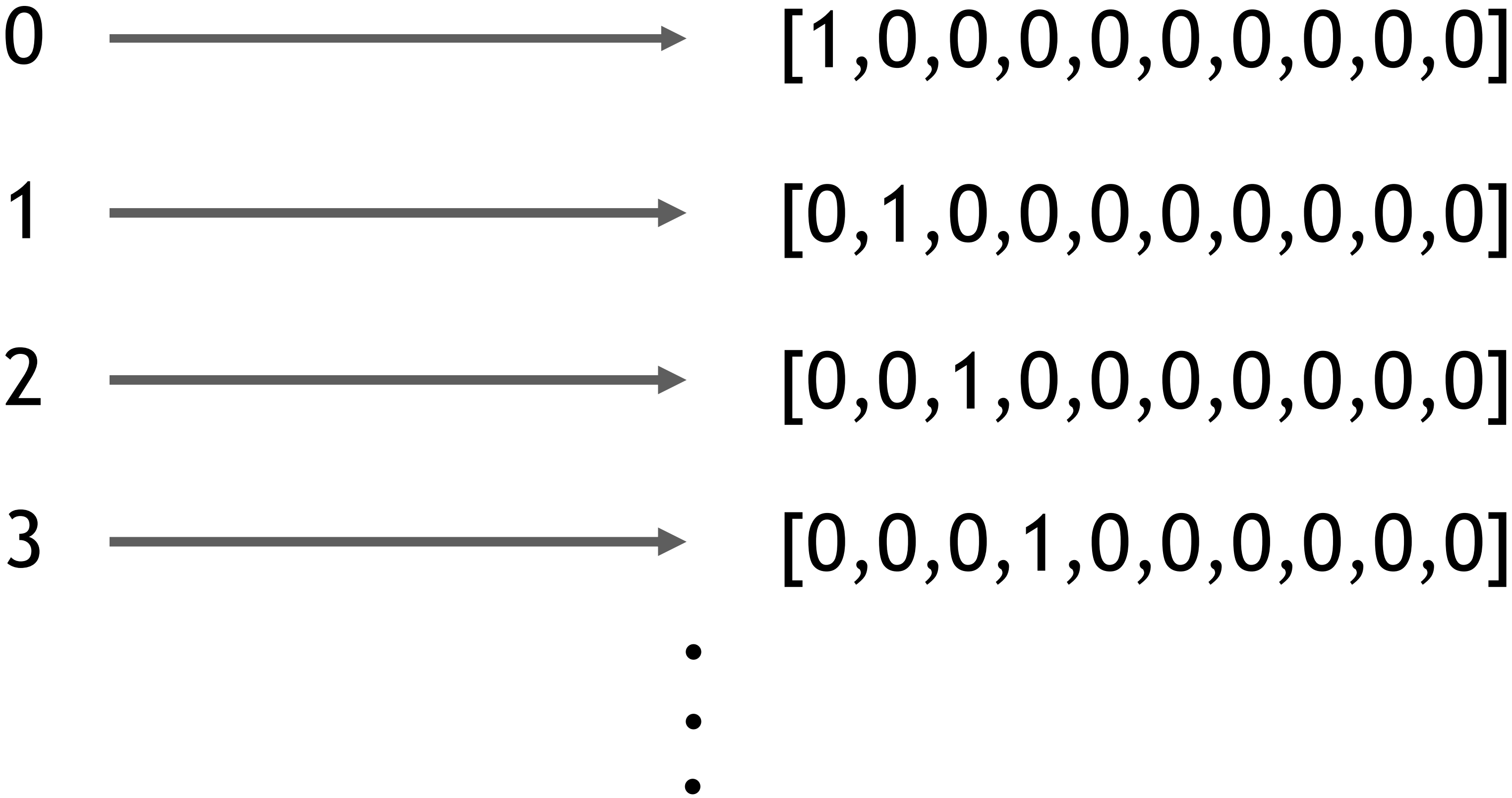


[0,0,0,24,75,184,185,78,32,55,0,0,0...]

Source : developer.nvidia.com/deep-leaning-course

DATA PREPARATION

Targets as categories



0 → [1,0,0,0,0,0,0,0,0,0]

1 → [0,1,0,0,0,0,0,0,0,0]

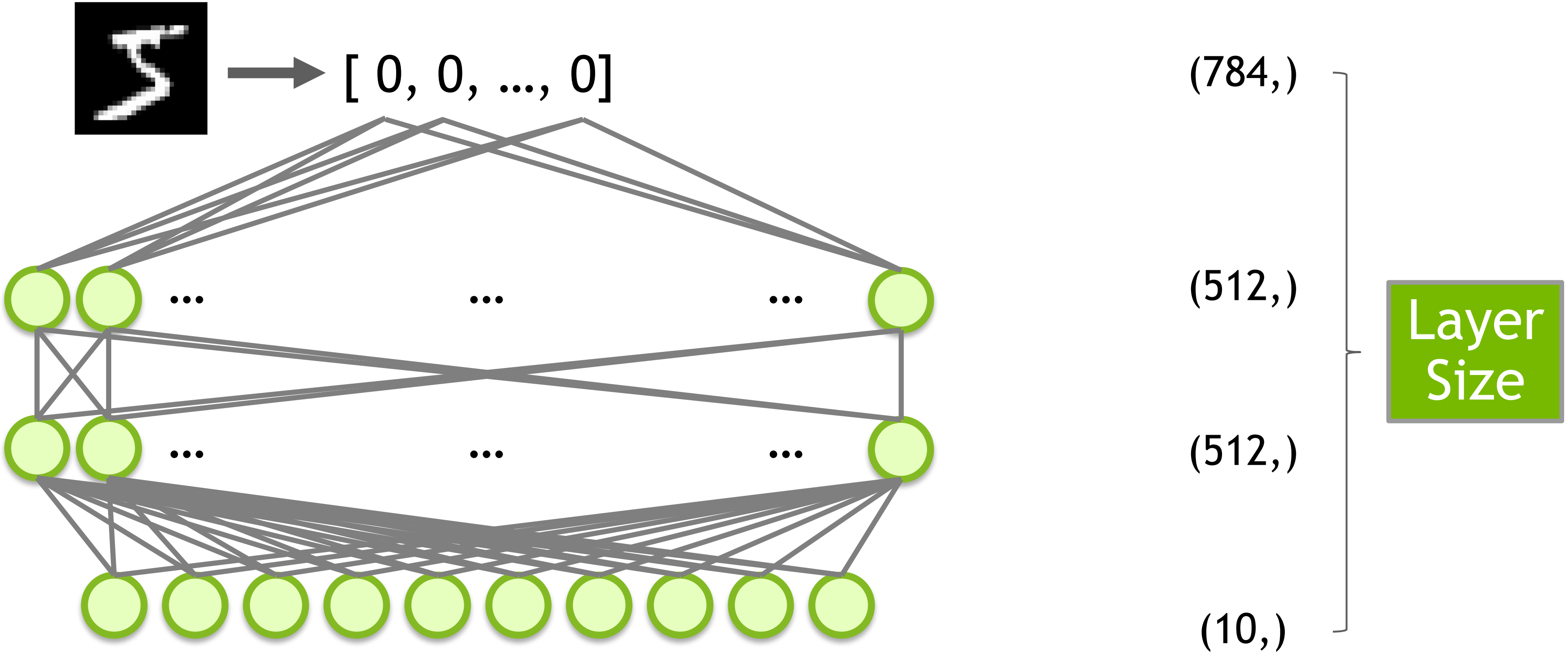
2 → [0,0,1,0,0,0,0,0,0,0]

3 → [0,0,0,1,0,0,0,0,0,0]

⋮

Source : developer.nvidia.com/deep-leaning-course

AN UNTRAINED MODEL

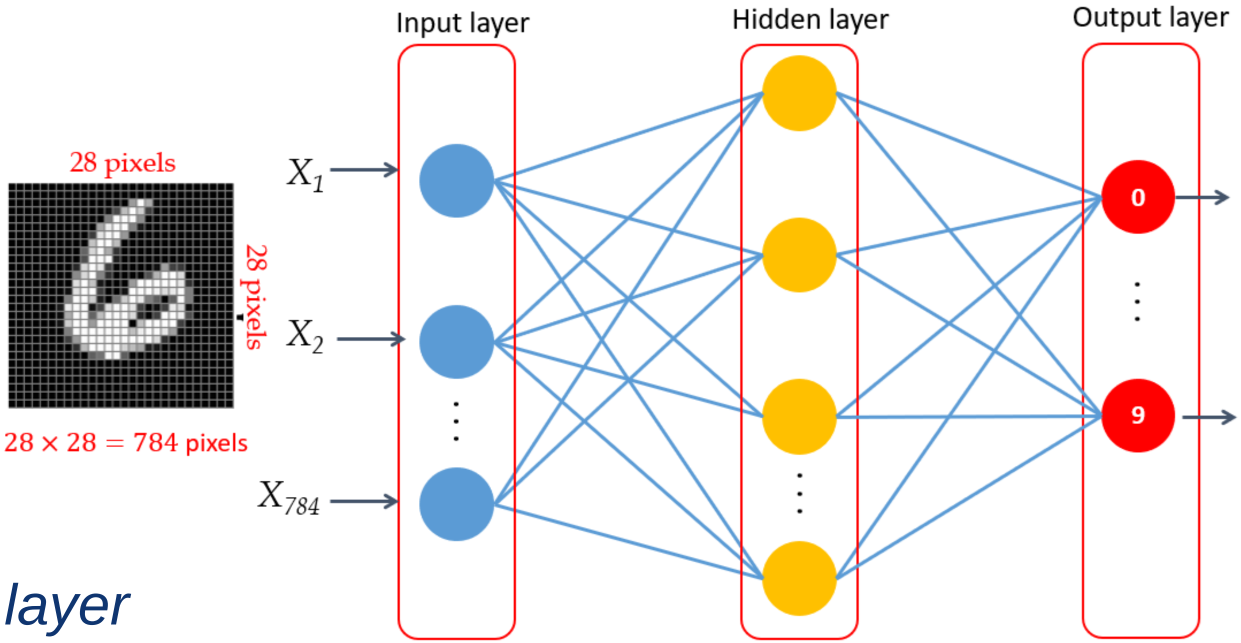


Source : developer.nvidia.com/deep-leaning-course

Desain arsitektur ANN

Penentuan jumlah *neuron* pada *input layer*

- Jumlah neuron sesuai dengan jumlah fitur pada data input



Penentuan jumlah *neuron* pada *output layer*

- Jumlah neuron sesuai dengan permasalahan
- Pada permasalahan klasifikasi biner dan regresi bisa menggunakan hanya satu *neuron*
- Pada permasalahan klasifikasi *multiclass* menggunakan jumlah *neuron* sesuai jumlah label kelasnya, misalnya: 10 neuron pada pengenalan angka

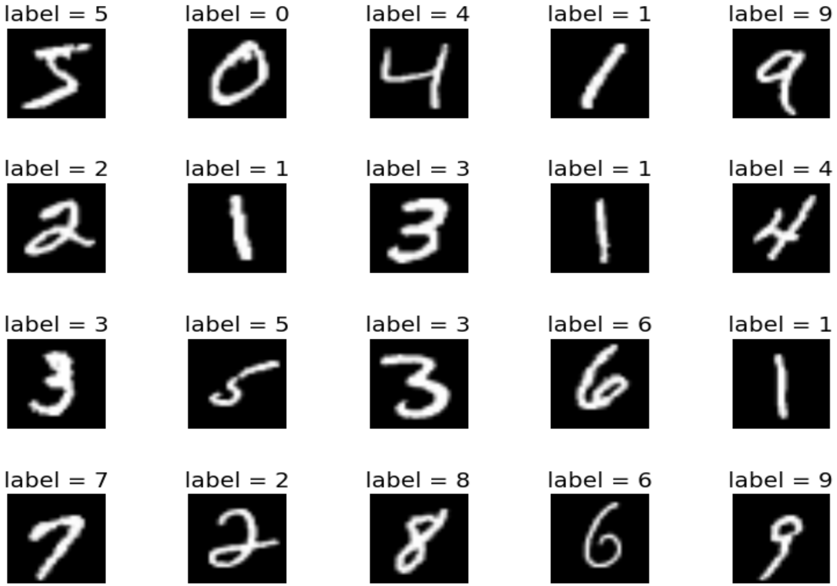
Source : Microcredential Associate Data Science – Dikti 2021

Contoh implementasi arsitektur *fully connected layer* pada pengenalan angka

Dataset MNIST *Handwritten Digit* dibagi menjadi 3:

- 55,000 training data
- 10,000 test data
- 5,000 validation data

Setiap citra berukuran 28 × 28 pixels dan label kelas diubah menjadi *one hot encoded*



0	[1 0 0 0 0 0 0 0 0 0]
1	[0 1 0 0 0 0 0 0 0 0]
2	[0 0 1 0 0 0 0 0 0 0]
3	[0 0 0 1 0 0 0 0 0 0]
4	[0 0 0 0 1 0 0 0 0 0]
5	[0 0 0 0 0 1 0 0 0 0]
6	[0 0 0 0 0 0 1 0 0 0]
7	[0 0 0 0 0 0 0 1 0 0]

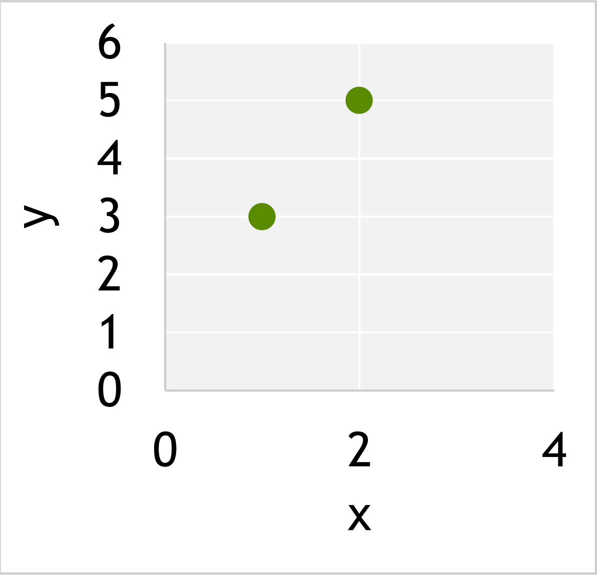
Source : Microcredential Associate Data Science – Dikti 2021

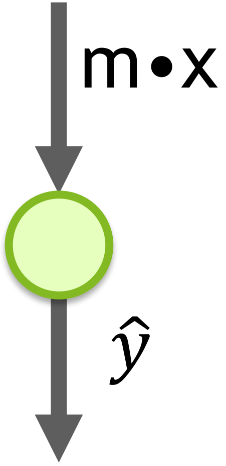
A SIMPLER MODEL

A SIMPLER MODEL

$$y = mx + b$$

x	y
1	3
2	5





$$m = ?$$

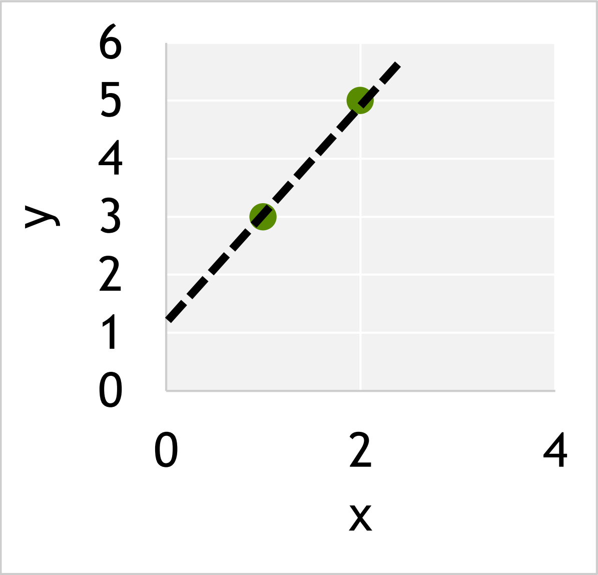
$$\mathrm{b} = ?$$

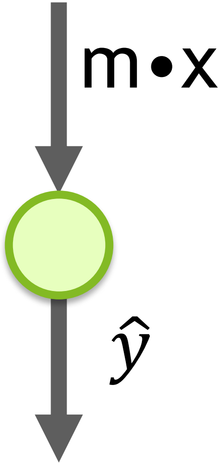
Source : developer.nvidia.com/deep-leaning-course

A SIMPLER MODEL

$$y = mx + b$$

x	y
1	3
2	5





$m = ?$

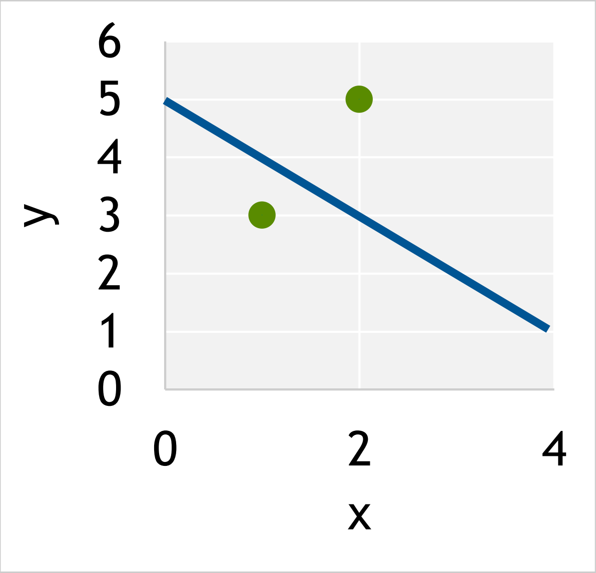
$b = ?$

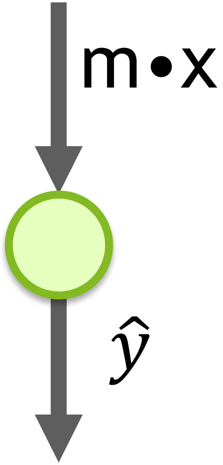
Source : developer.nvidia.com/deep-leaning-course

A SIMPLER MODEL

$$y = mx + b$$

x	y	$\hat{y}$
1	3	4
2	5	3





Start Random

$$m = -1$$

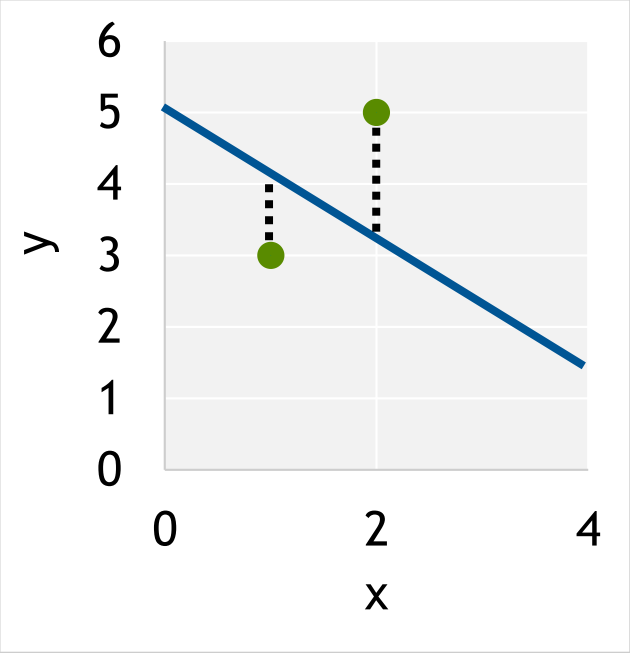
$$b = 5$$

Source : developer.nvidia.com/deep-leaning-course

A SIMPLER MODEL

$$y = mx + b$$

x	y	$\hat{y}$	err^2
1	3	4	1
2	5	3	4
		MSE =	2.5
		RMSE =	1.6



$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

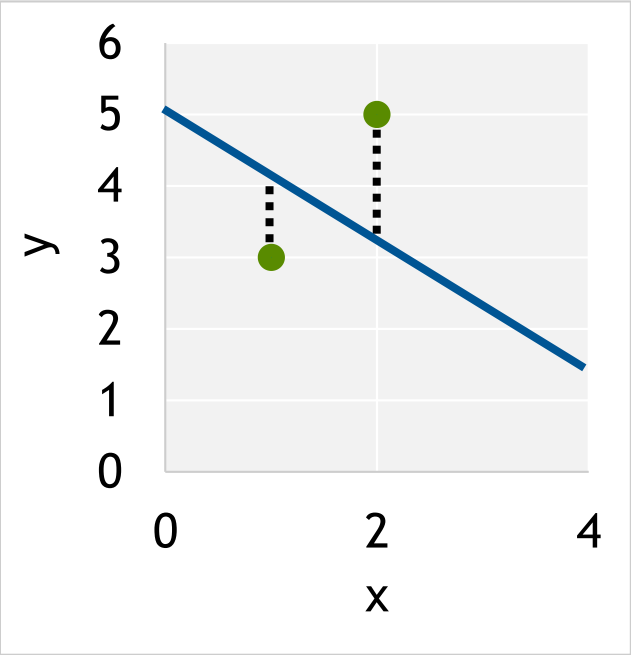
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

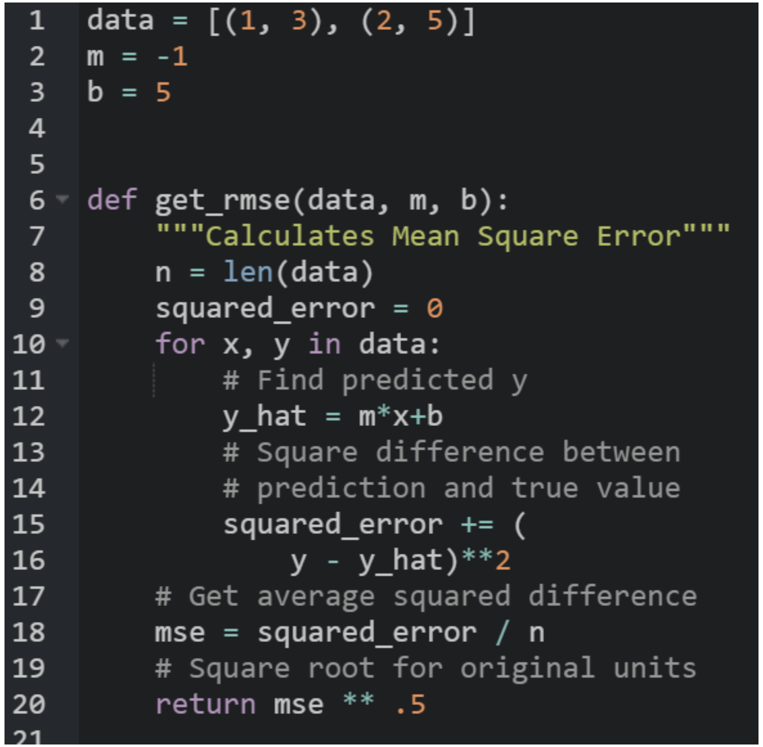
Source : developer.nvidia.com/deep-leaning-course

A SIMPLER MODEL

$$y = mx + b$$

x	y	$\hat{y}$	err^2
1	3	4	1
2	5	3	4
		MSE =	2.5
		RMSE =	1.6

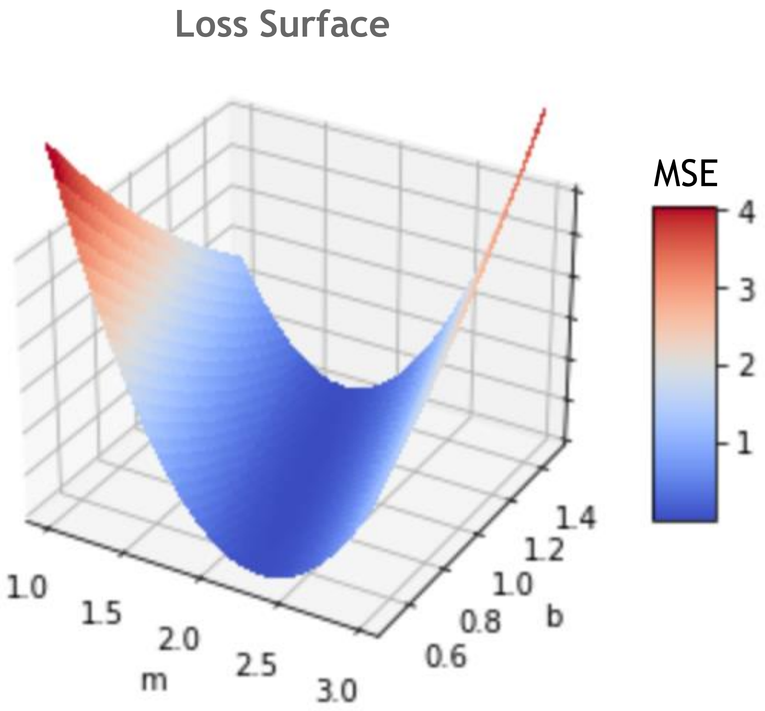




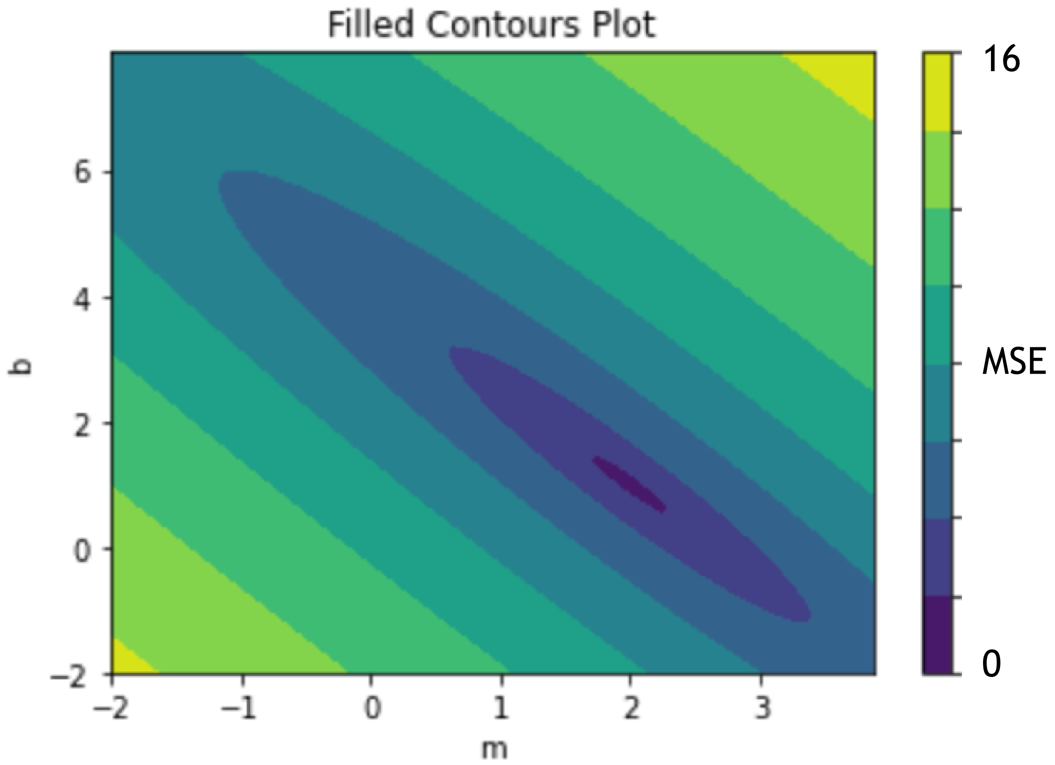
```python
data = [(1, 3), (2, 5)]
m = -1
b = 5


def get_rmse(data, m, b):
    """Calculates Mean Square Error"""
    n = len(data)
    squared_error = 0
    for x, y in data:
        # Find predicted y
        y_hat = m*x+b
        # Square difference between
        # prediction and true value
        squared_error += (
            y - y_hat)**2
    # Get average squared difference
    mse = squared_error / n
    # Square root for original units
    return mse ** .5
```

Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE



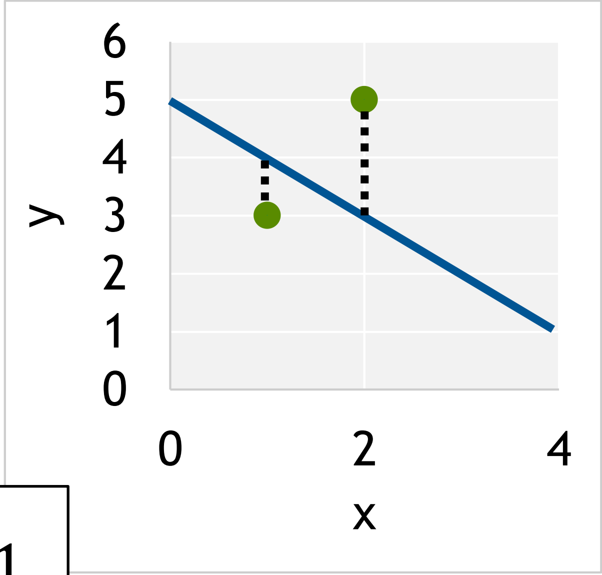
Loss Surface



Filled Contours Plot

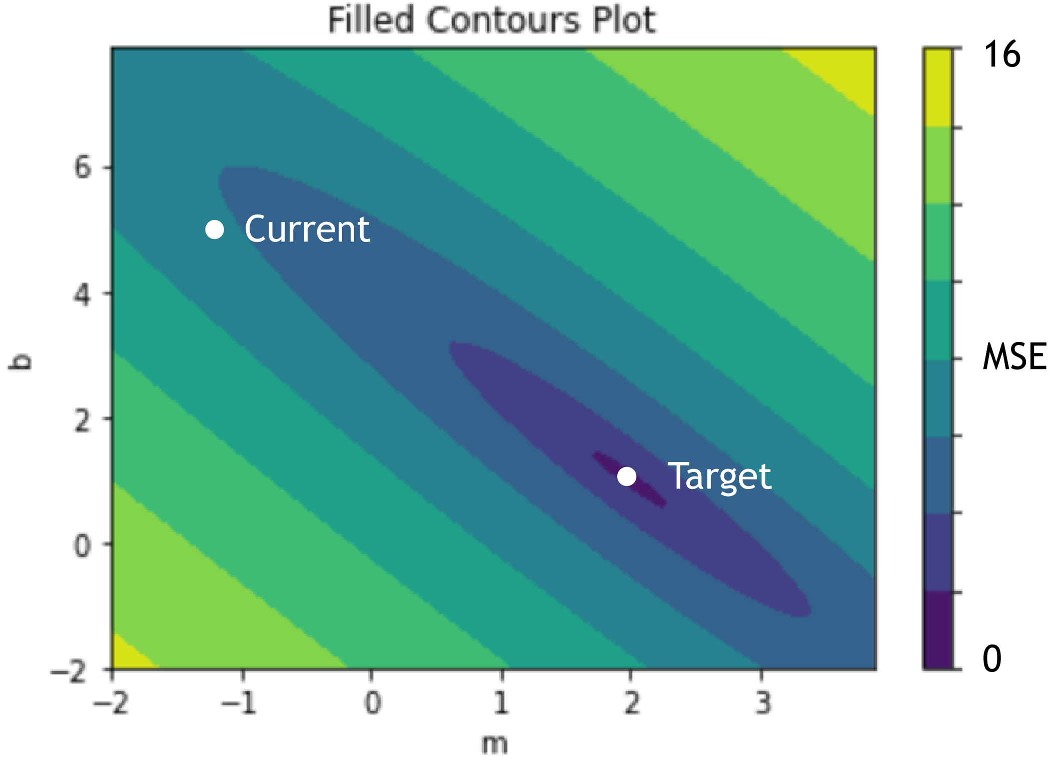
Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE



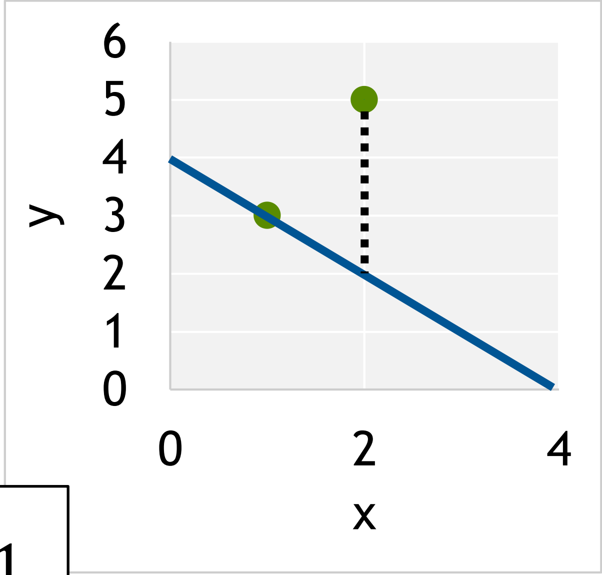
$$m = -1$$

$$\mathrm{b} = 5$$



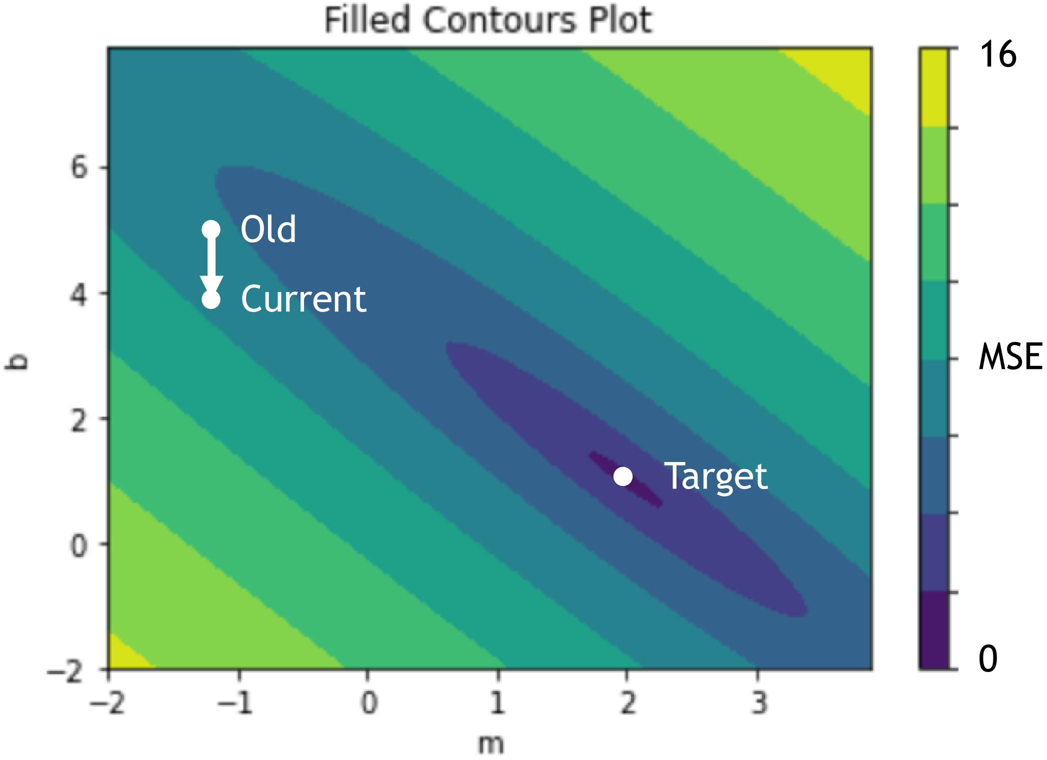
Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE



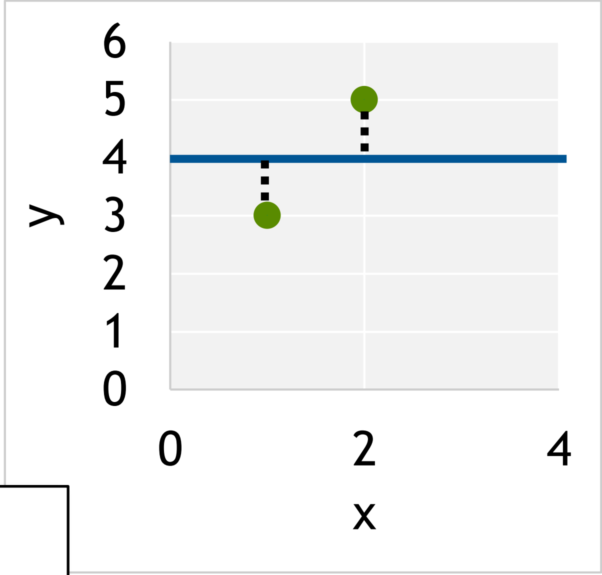
$m = -1$

$b = 4$



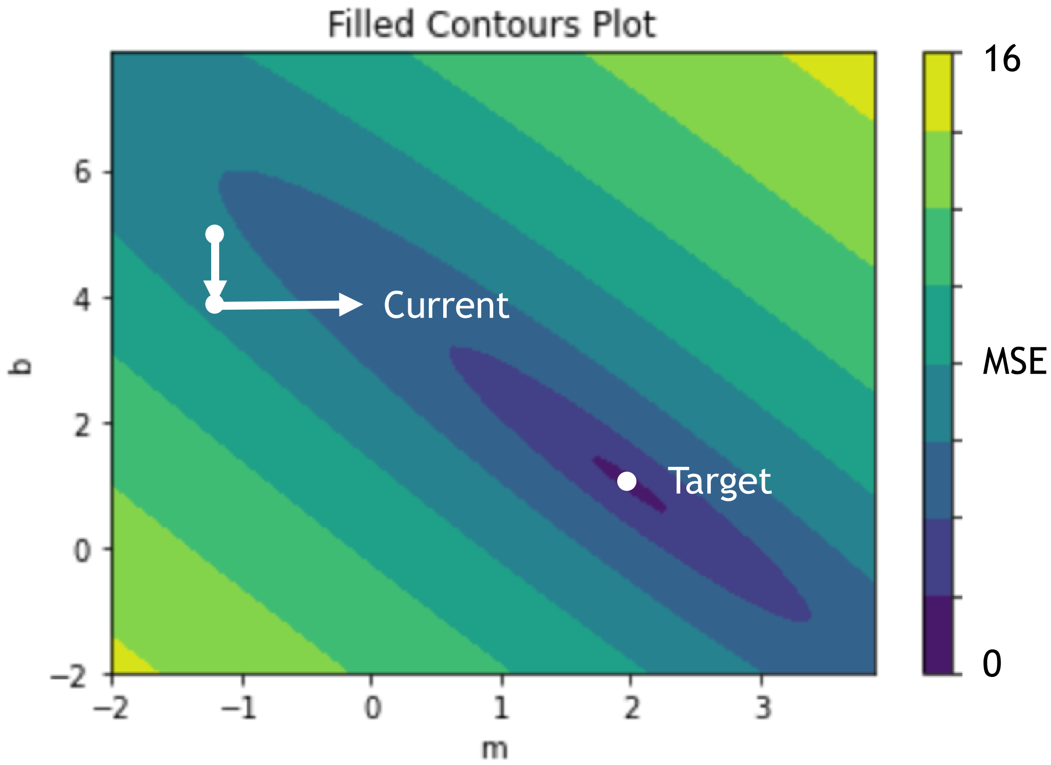
Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE



$$m = 0$$

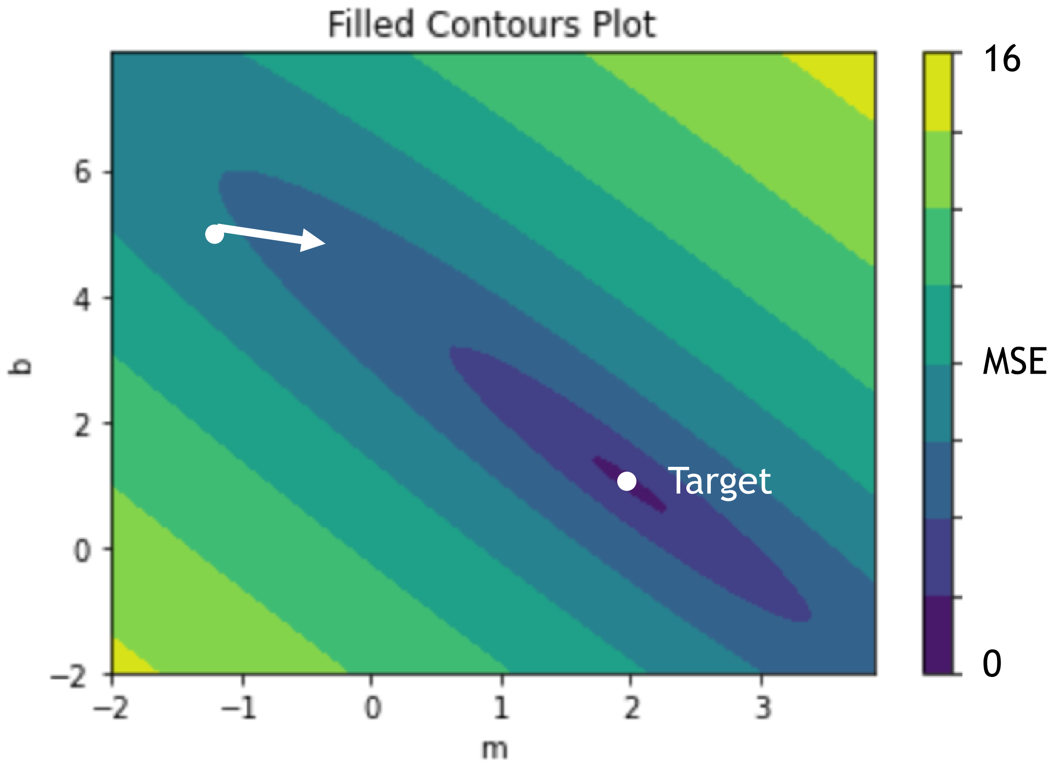
$$b = 4$$



Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE

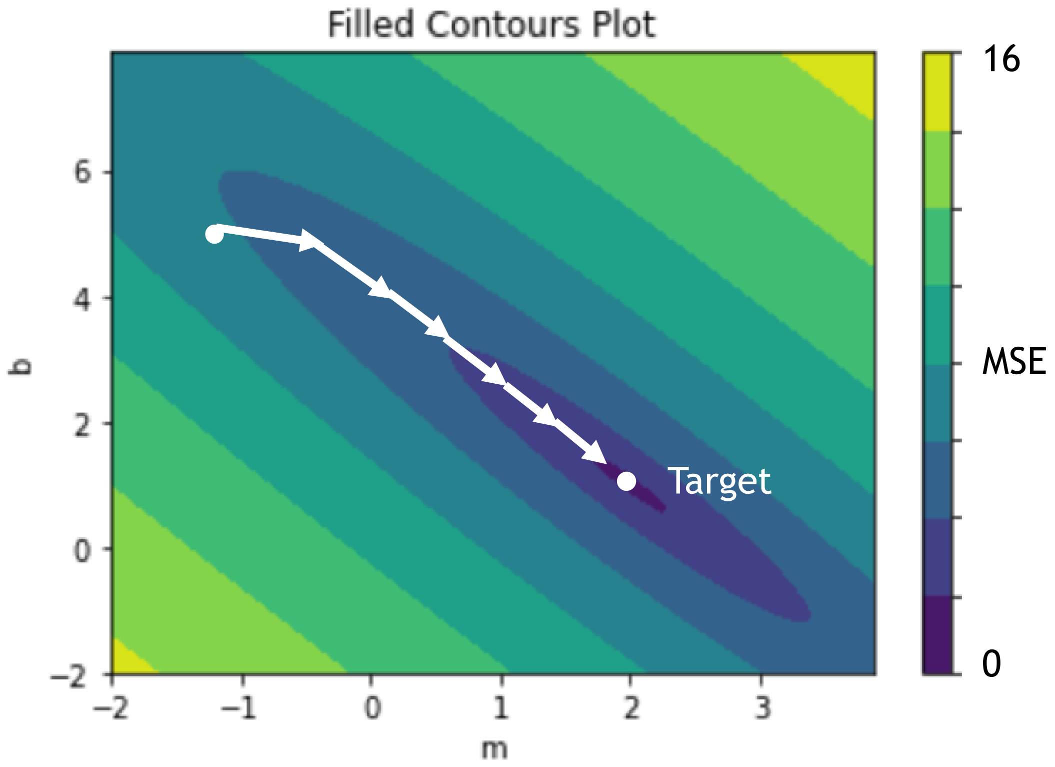
•The Gradient	Which direction loss decreases the most
λ: The learning rate	How far to travel
Epoch	A model update with the full dataset
Batch	A sample of the full dataset
Step	An update to the weight parameters



Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE

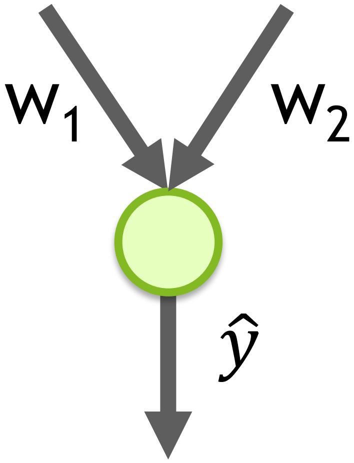
•The Gradient	Which direction loss decreases the most
λ: The learning rate	How far to travel
Epoch	A model update with the full dataset
Batch	A sample of the full dataset
Step	An update to the weight parameters



Source : developer.nvidia.com/deep-leaning-course

FROM NEURON TO NETWORK

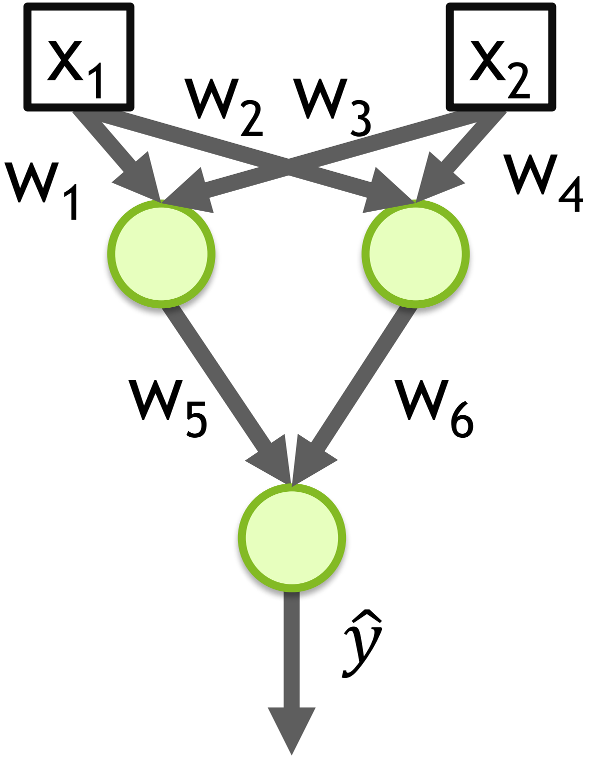
BUILDING A NETWORK



- Scales to more inputs

Source : developer.nvidia.com/deep-leaning-course

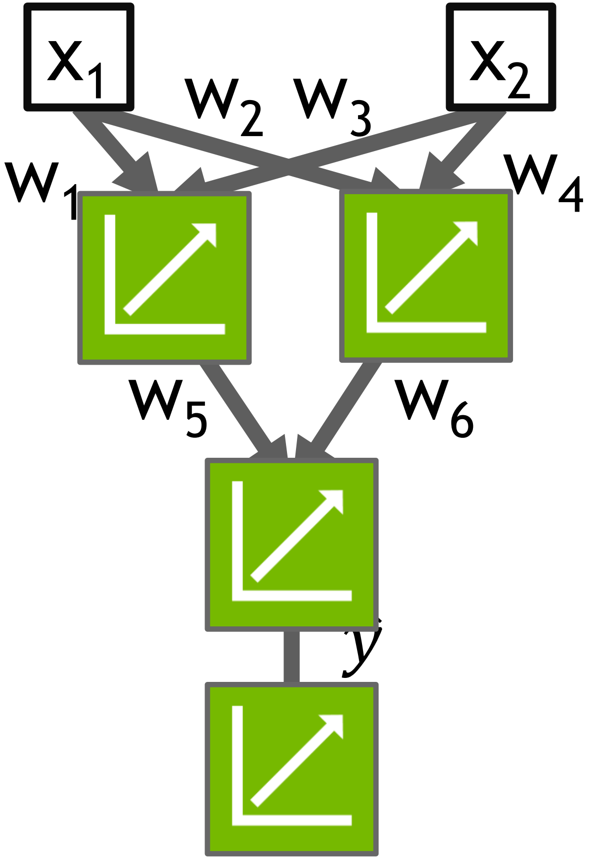
BUILDING A NETWORK



- Scales to more inputs
- Can chain neurons

Source : developer.nvidia.com/deep-leaning-course

BUILDING A NETWORK



- Scales to more inputs
- Can chain neurons
- If all regressions are linear, then output will also be a linear regression

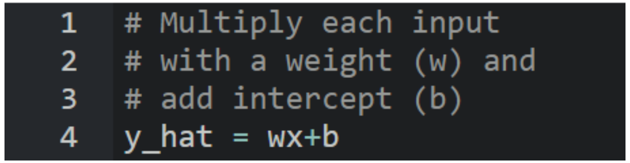
Source : developer.nvidia.com/deep-leaning-course

ACTIVATION FUNCTIONS

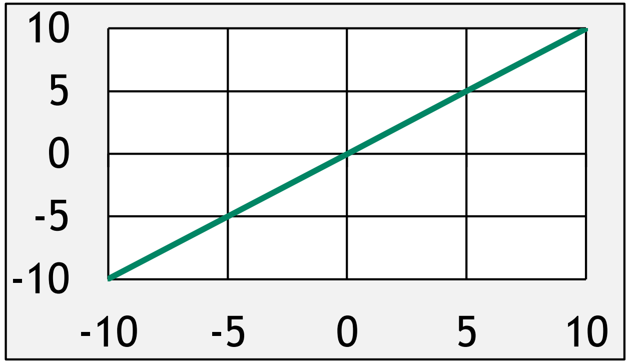
ACTIVATION FUNCTIONS

Linear

$$\hat{y} = wx + b$$

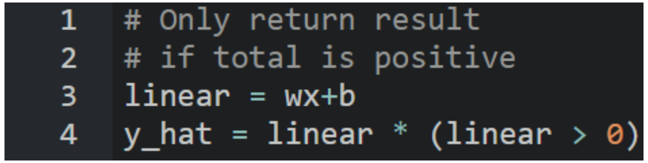


```
# Multiply each input
# with a weight (w) and
# add intercept (b)
y_hat = wx+b
```

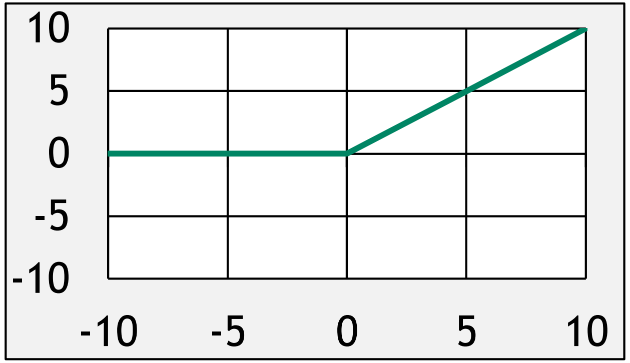


ReLU

$$\hat{y} = \begin{cases} wx + b \; if \; wx + b > 0 \\ 0 \;\; otherwise \end{cases}$$

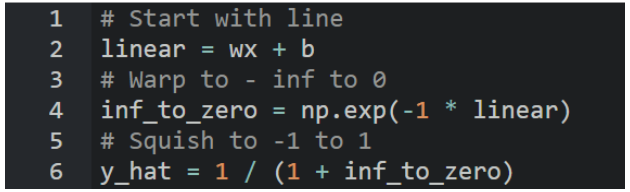


```
# Only return result
# if total is positive
linear = wx+b
y_hat = linear * (linear > 0)
```

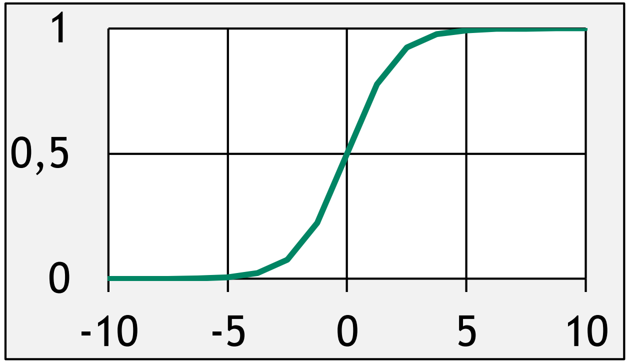


Sigmoid

$$\hat{y} = \frac{1}{1 + e^{-(wx+b)}}$$



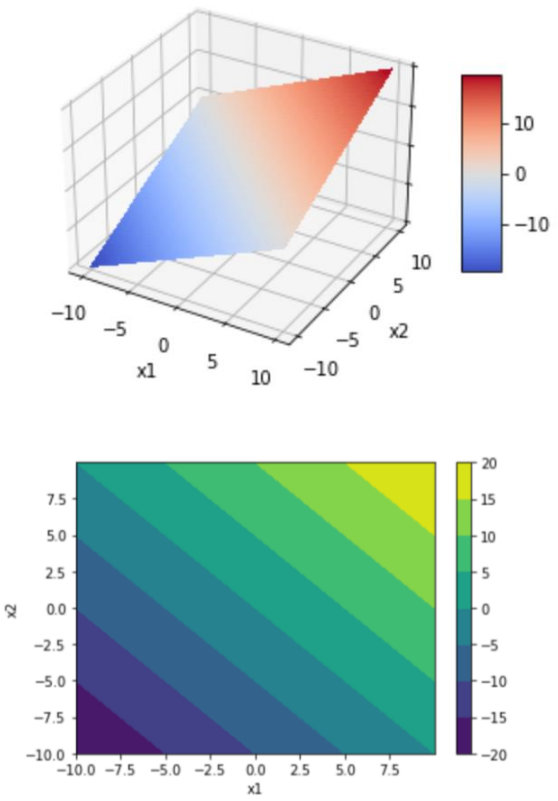
```
# Start with line
linear = wx + b
# Warp to - inf to 0
inf_to_zero = np.exp(-1 * linear)
# Squish to -1 to 1
y_hat = 1 / (1 + inf_to_zero)
```



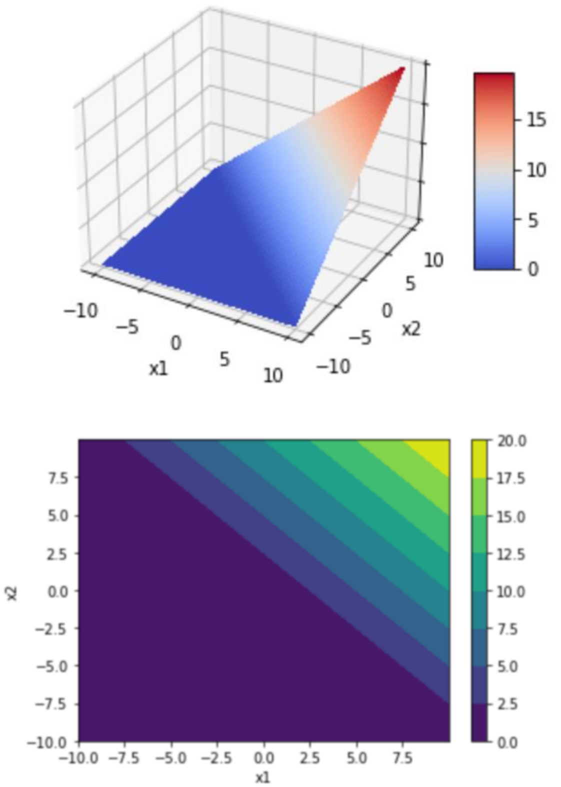
Source : developer.nvidia.com/deep-leaning-course

ACTIVATION FUNCTIONS

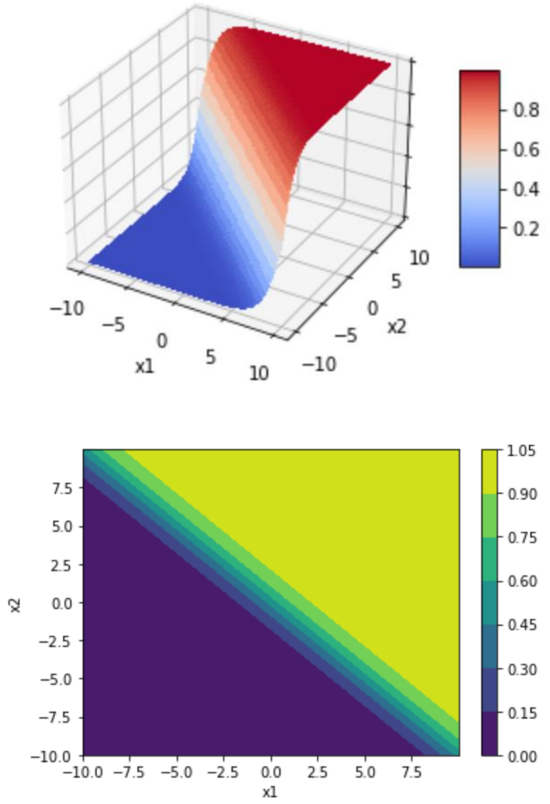
Linear



ReLU

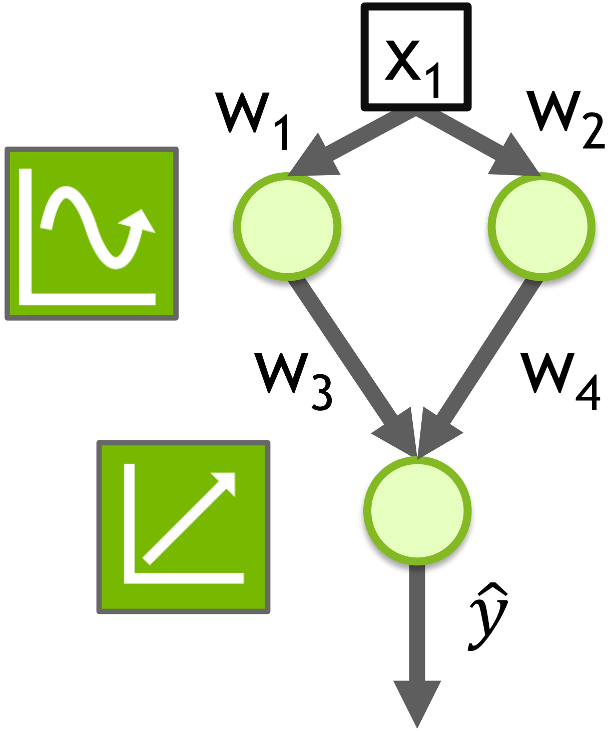


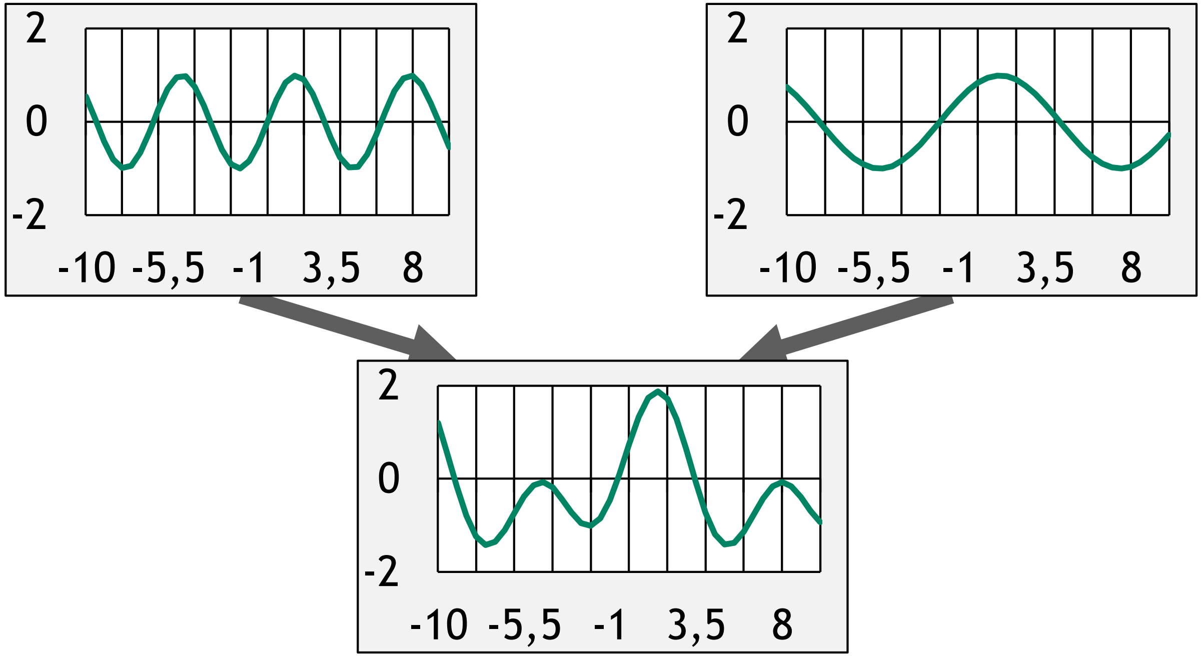
Sigmoid



Source : developer.nvidia.com/deep-leaning-course

ACTIVATION FUNCTIONS



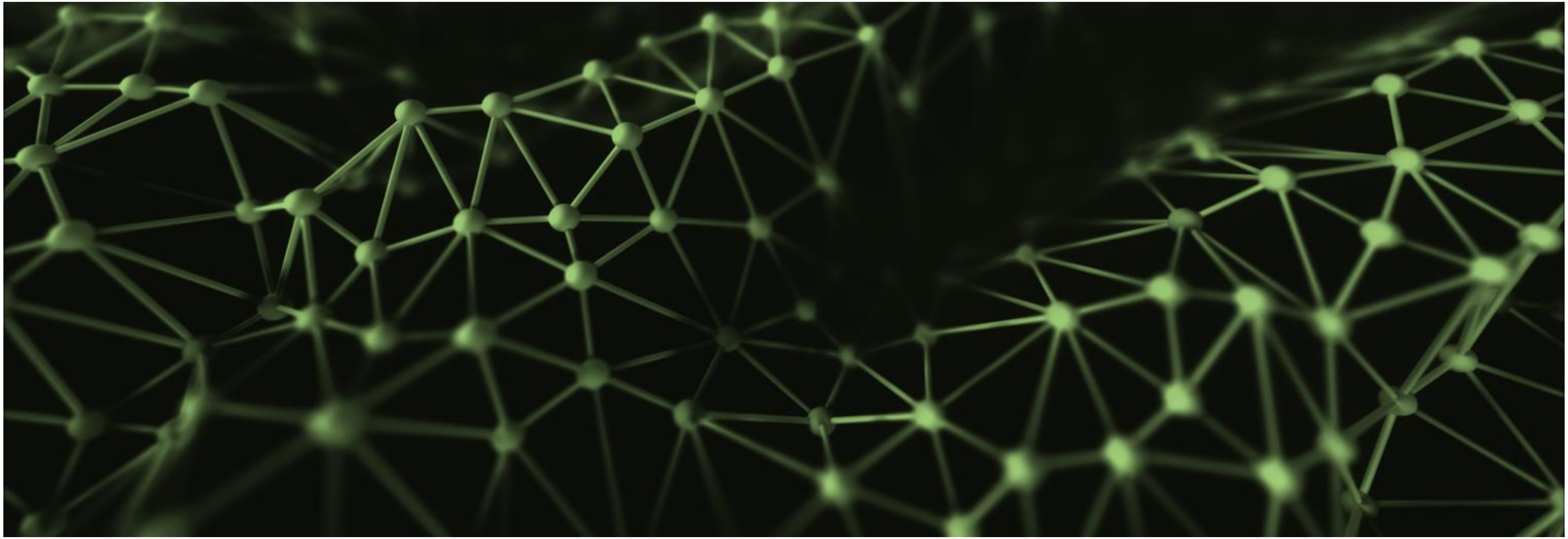


Source : developer.nvidia.com/deep-leaning-course

OVERFITTING

OVERFITTING

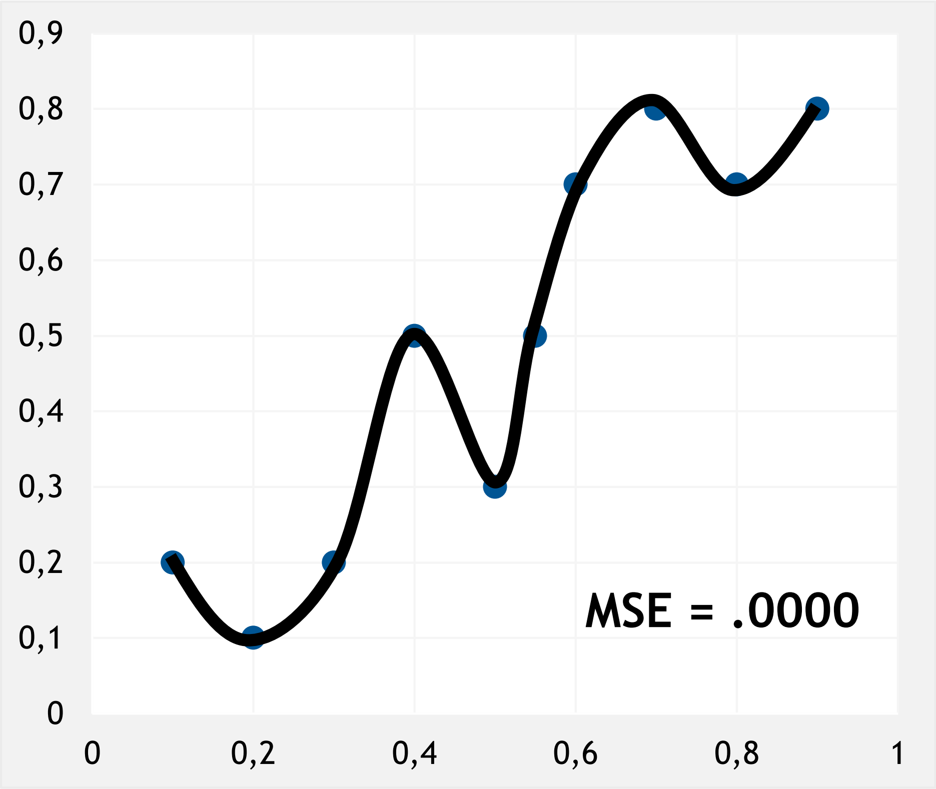
Why not have a super large neural network?

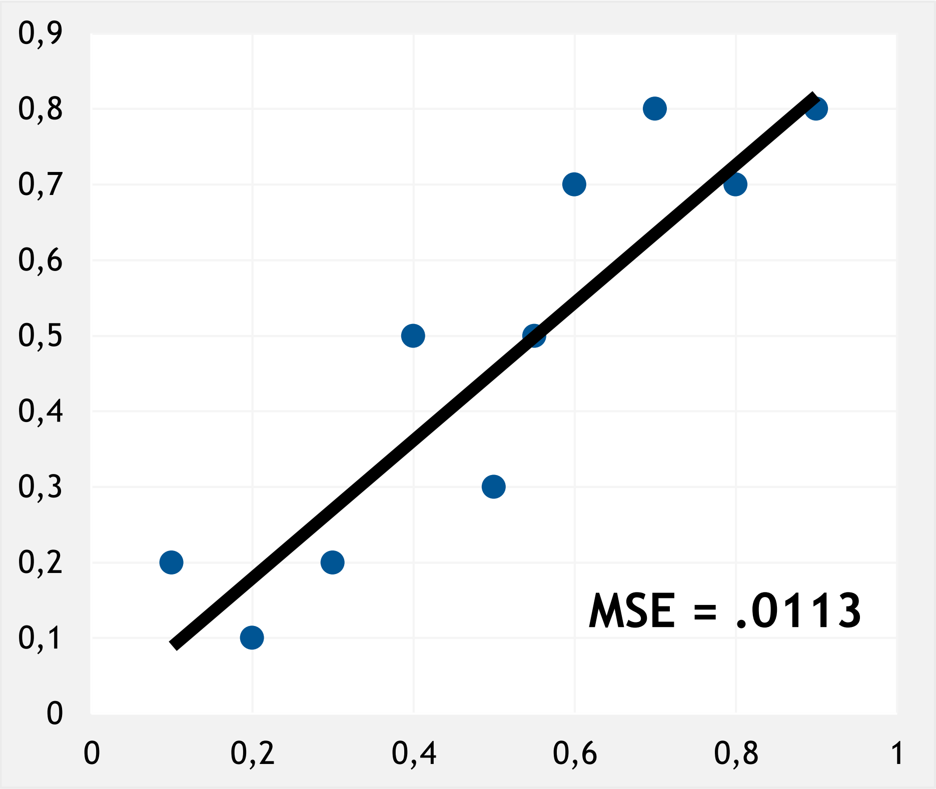


Source : developer.nvidia.com/deep-leaning-course

OVERFITTING

Which Trendline is Better?

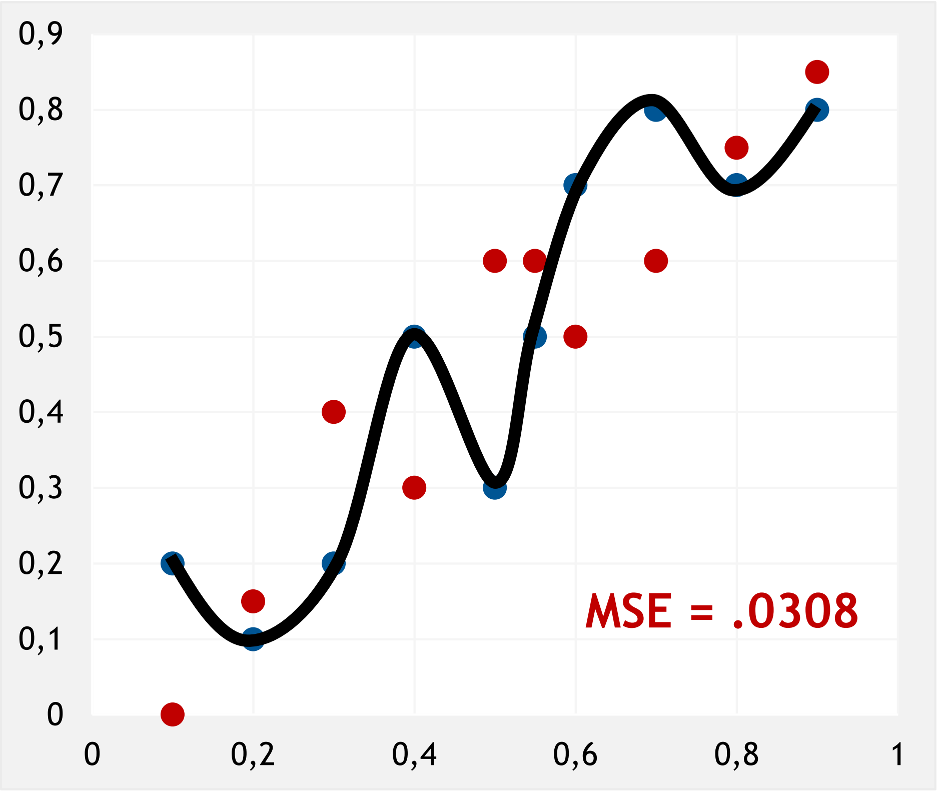


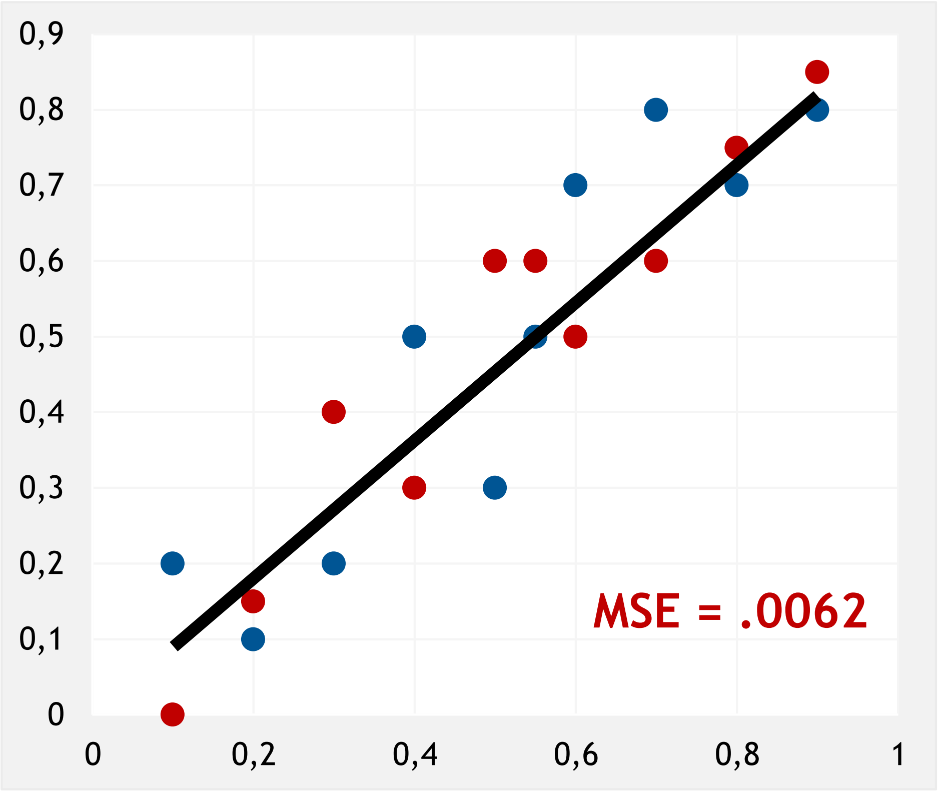


Source : developer.nvidia.com/deep-leaning-course

OVERFITTING

Which Trendline is Better?





Source : developer.nvidia.com/deep-leaning-course

TRAINING VS VALIDATION DATA

Avoid memorization

Training data

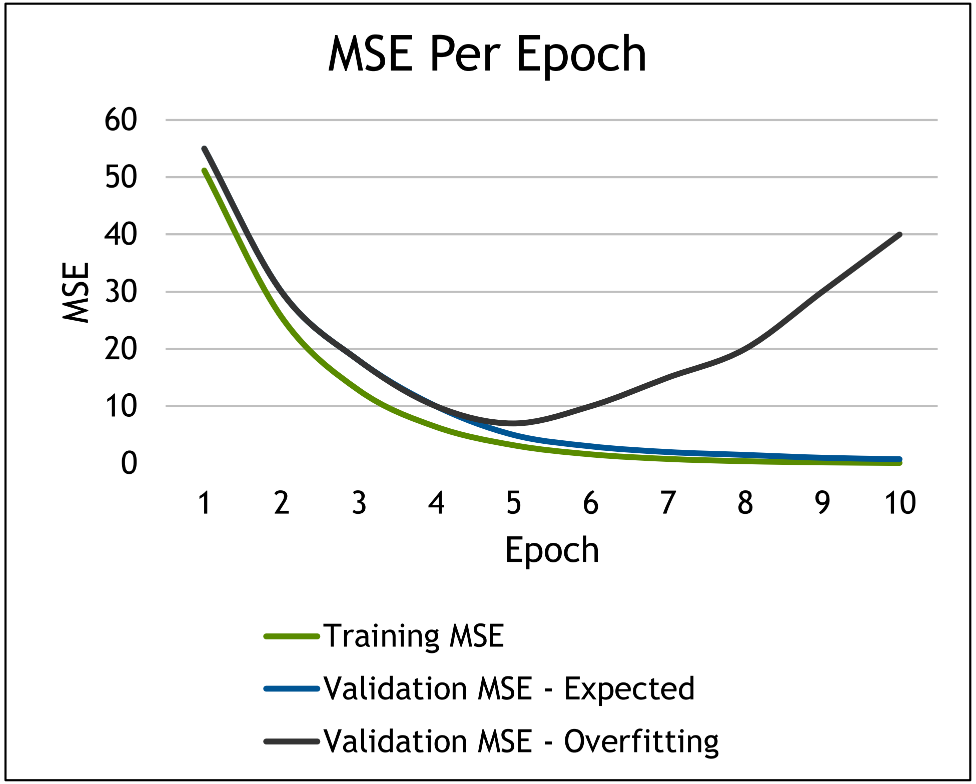
- Core dataset for the model to learn on

Validation data

- New data for model to see if it truly understands (can generalize)

Overfitting

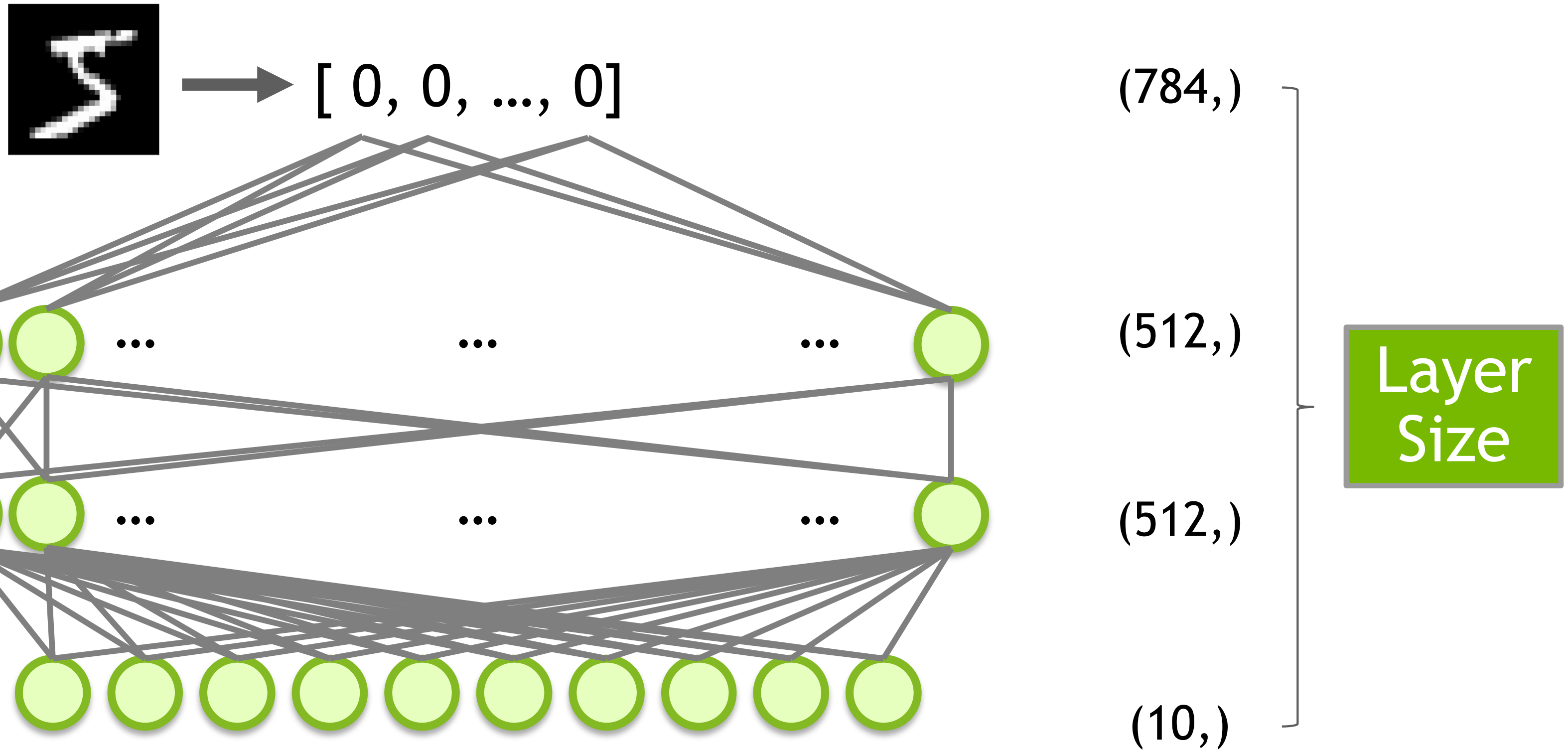
- When model performs well on the training data, but not the validation data (evidence of memorization)
- Ideally the accuracy and loss should be similar between both datasets



Source : developer.nvidia.com/deep-leaning-course

FROM REGRESSION TO CLASSIFICATION

AN MNIST MODEL



[0, 0, …, 0]

(784,)

(512,)

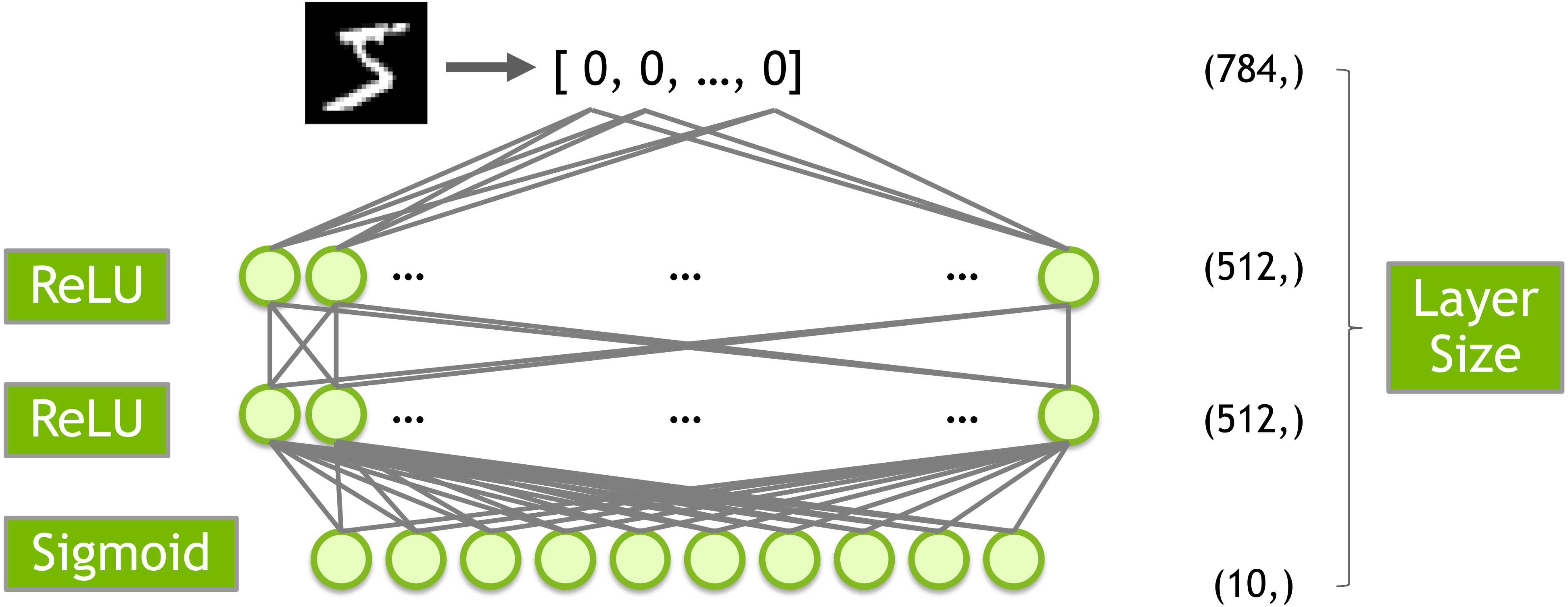
(512,)

(10,)

Layer Size

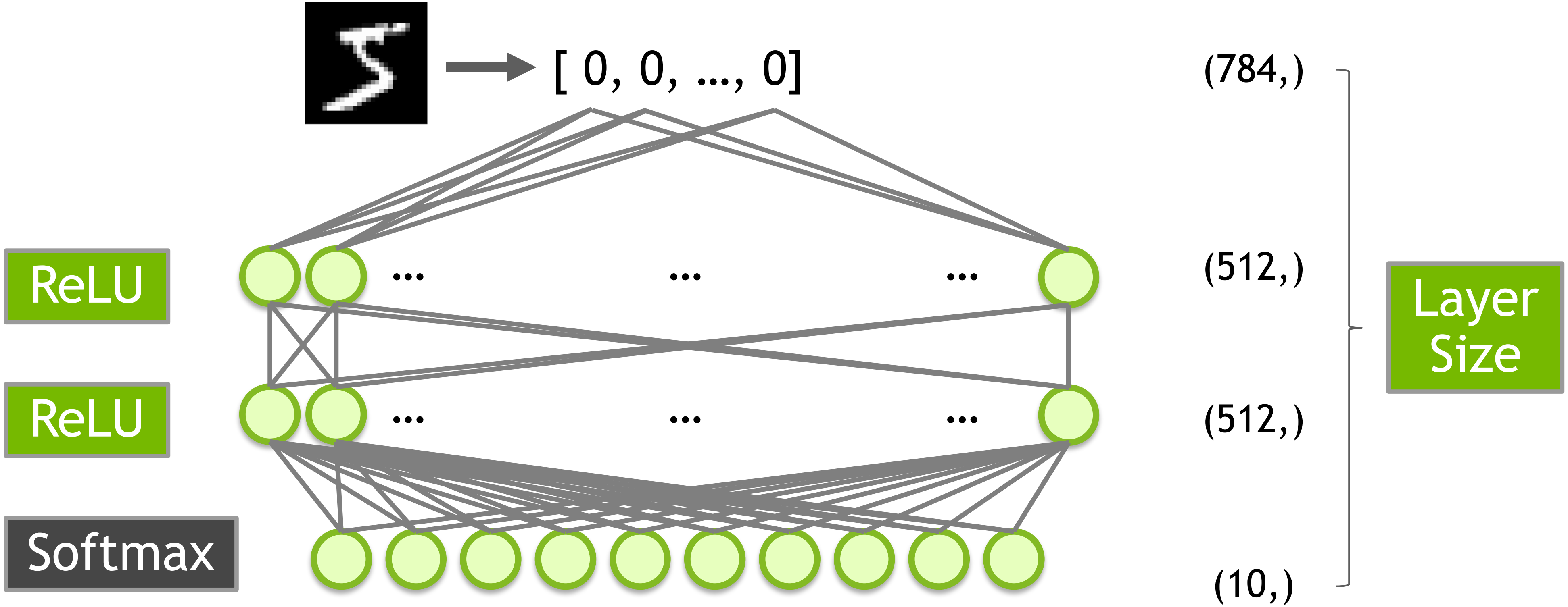
Source : developer.nvidia.com/deep-leaning-course

AN MNIST MODEL



Source : developer.nvidia.com/deep-leaning-course

AN MNIST MODEL



Source : developer.nvidia.com/deep-leaning-course

GRADIENT DESCENT

Learning From Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 = \frac{1}{n}\sum_{i=1}^{n}(y - (mx + b))^2$$

$$MSE = \frac{1}{2}((3 - (m(1) + b))^2 + (5 - (m(2) + b))^2)$$

$$\frac{\partial MSE}{\partial m} = 5m + 3b - 13 \qquad \frac{\partial MSE}{\partial b} = 3m + 2b - 8$$

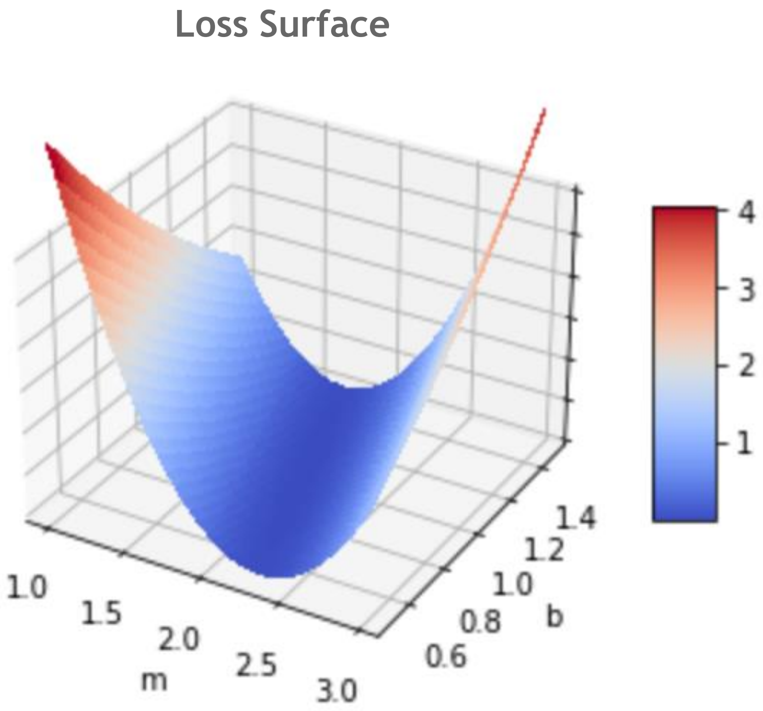
$$\frac{\partial MSE}{\partial m} = -3 \qquad \frac{\partial MSE}{\partial b} = -1$$

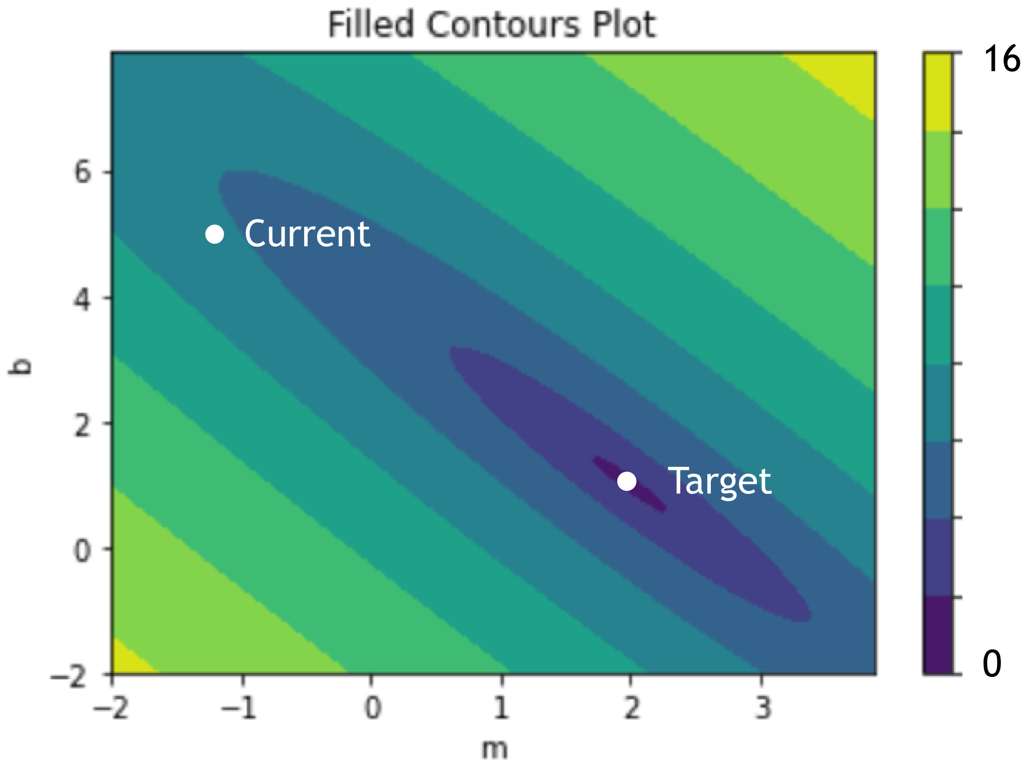
$m = -1$

b $= 5$

Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE

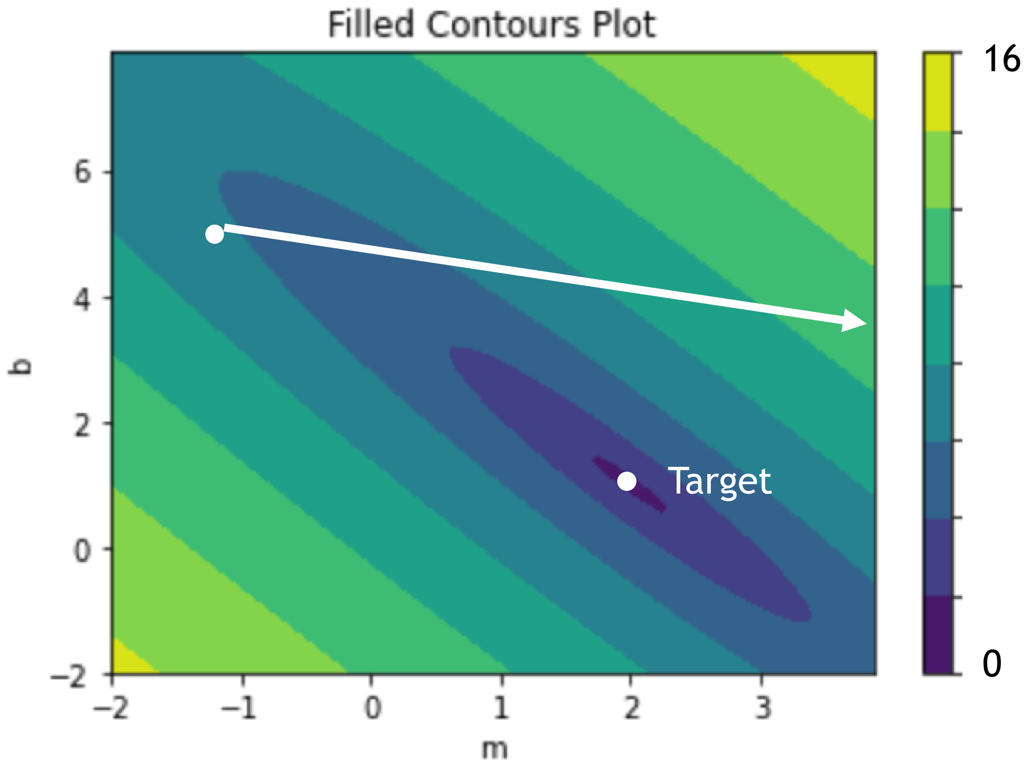




Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$



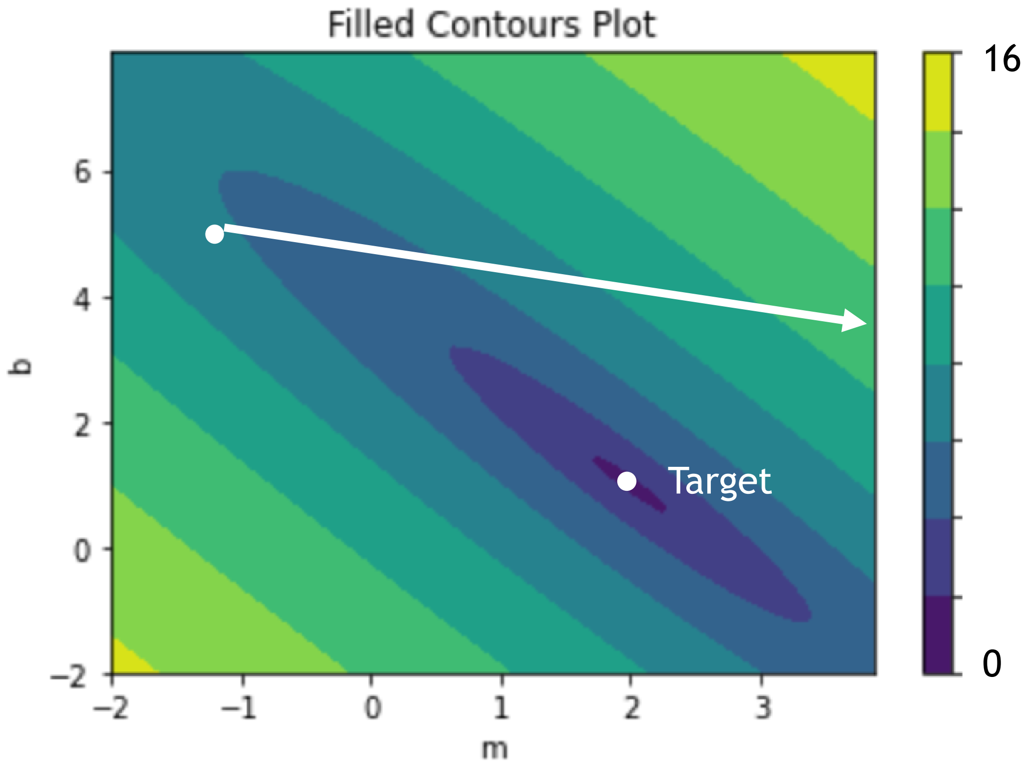
Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

$$\mathrm{m} := \mathrm{m} \ - \lambda \frac{\partial MSE}{\partial m}$$

$$\mathrm{b} \ := b \ - \lambda \frac{\partial MSE}{\partial b}$$



Source : developer.nvidia.com/deep-leaning-course

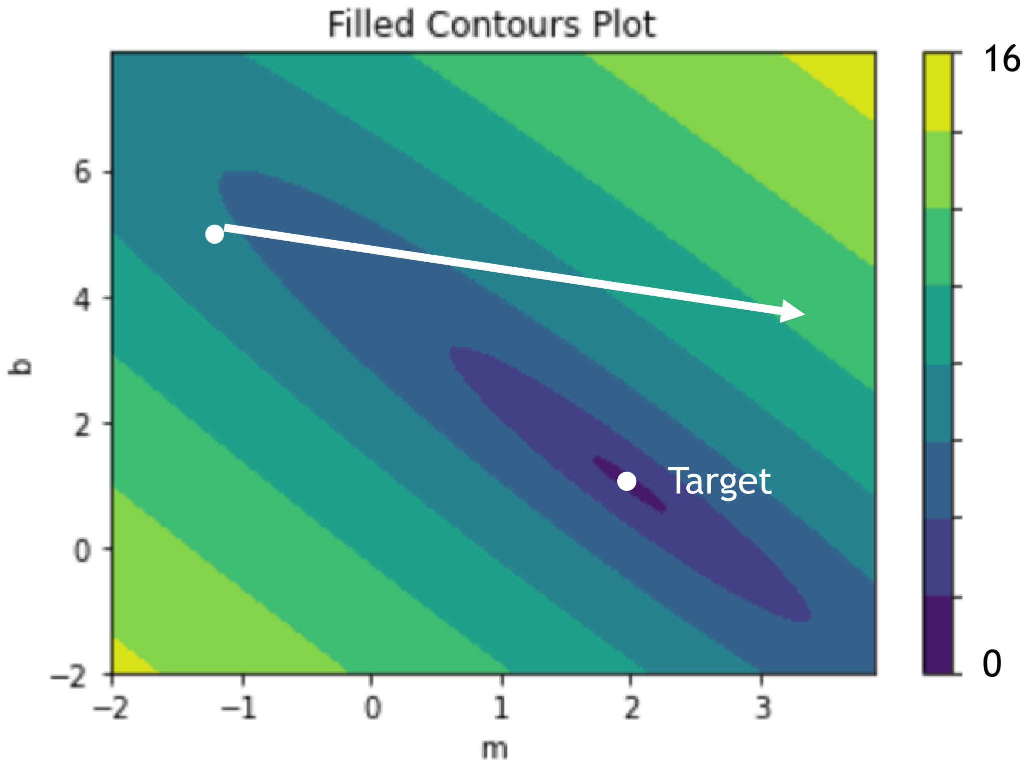
THE LOSS CURVE

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

$$\mathrm{m} := \mathrm{m} - \lambda \frac{\partial MSE}{\partial m}$$

$$\lambda = .6$$

$$\mathrm{b} := b - \lambda \frac{\partial MSE}{\partial b}$$



Source : developer.nvidia.com/deep-leaning-course

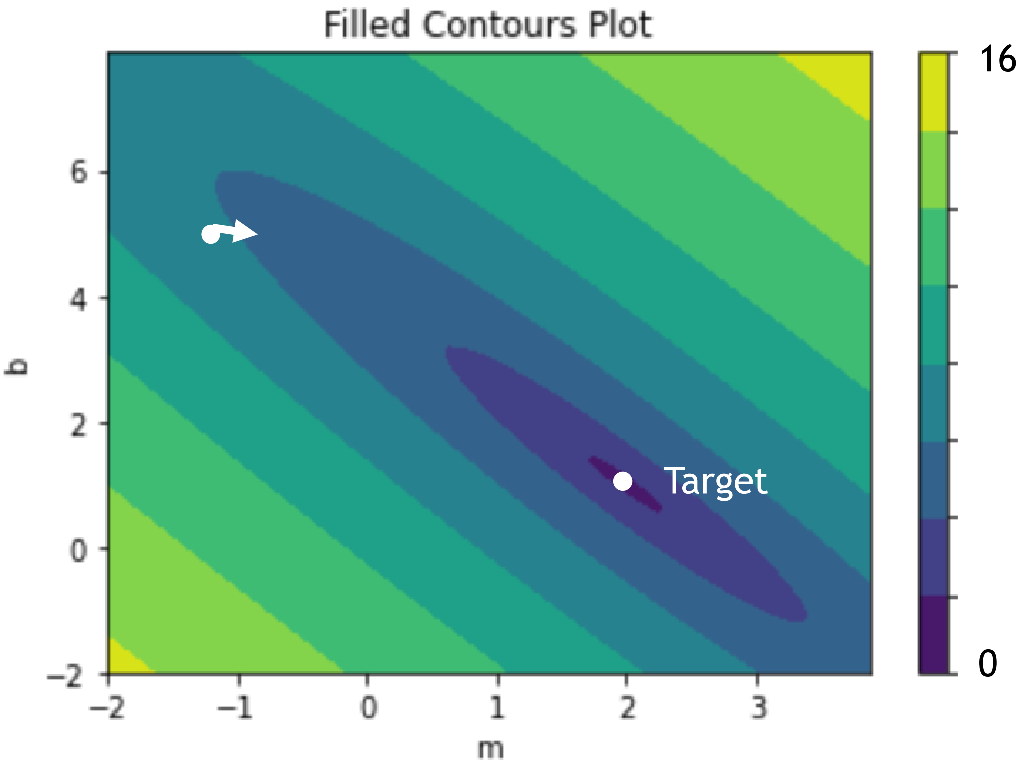
THE LOSS CURVE

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

$$\mathrm{m} := \mathrm{m} - \lambda \frac{\partial MSE}{\partial m}$$

$$\lambda = .005$$

$$\mathrm{b} := b - \lambda \frac{\partial MSE}{\partial b}$$



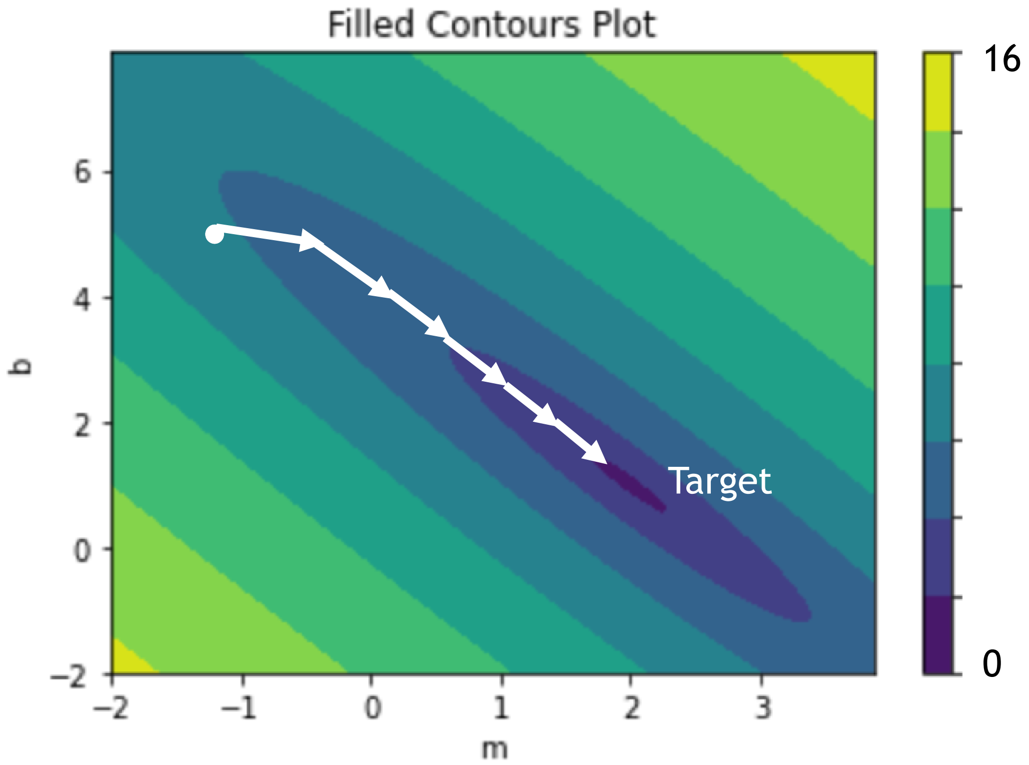
Source : developer.nvidia.com/deep-leaning-course

THE LOSS CURVE

$\lambda = .1$

$m := -1 + 7\,\lambda = -0.3$

$b := 5 + 3\,\lambda = 4.7$



Source : developer.nvidia.com/deep-leaning-course

SEMOGA BERMANFAAT