HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

Folien zur Vorlesung am 15.04.2025

3D Computer Vision

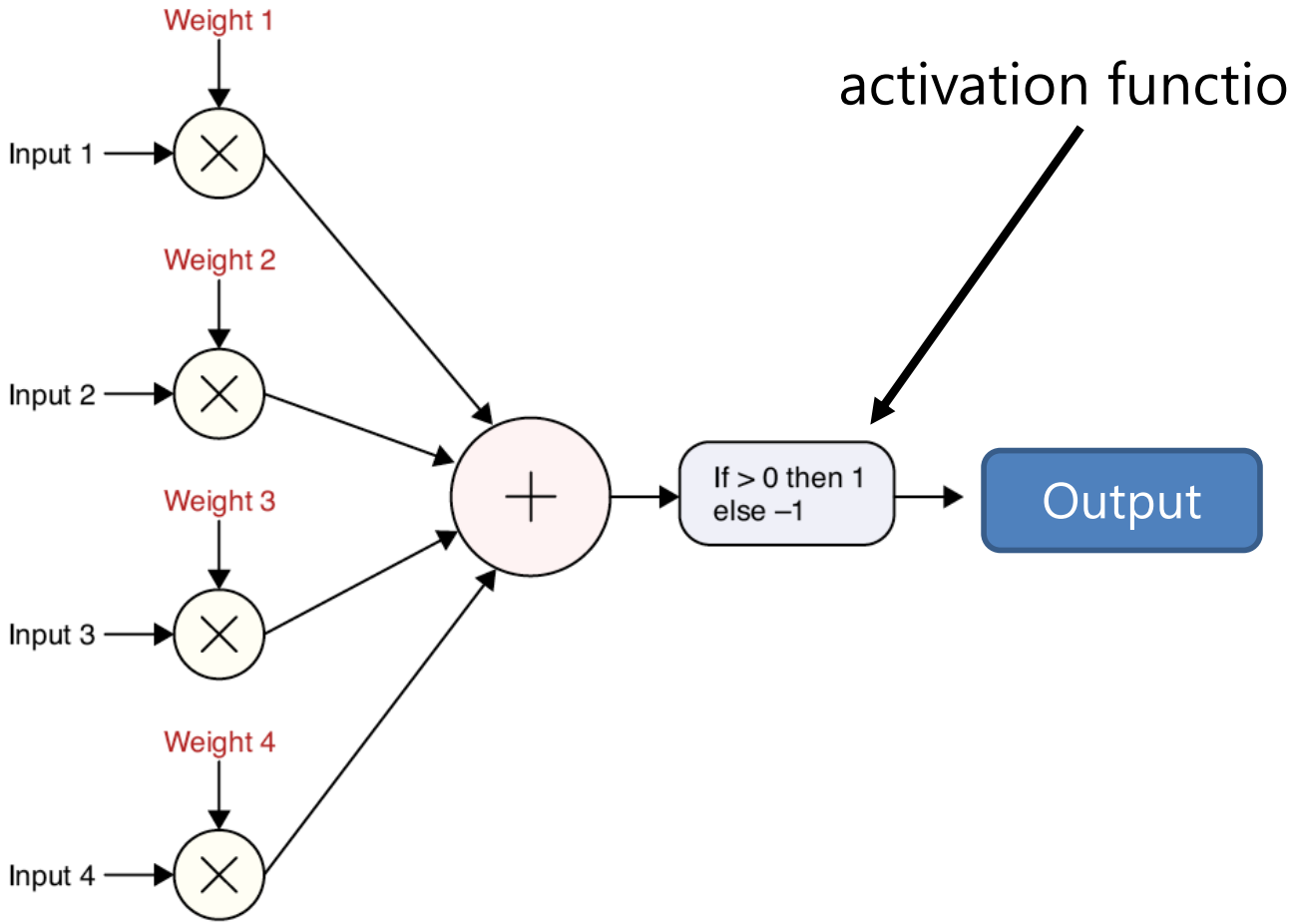
DEPTH FROM *A SINGLE IMAGE* USING DEEP LEARNING

What is Deep Learning? How does it work?

- Recommended videos:
  - Yann LeCun: Artificial intelligence, revealed
    - High level introduction with short animations (15 min)
  - Geoffrey Hinton: The Foundations of Deep Learning
    - A high level rather general and entertaining introduction (28 min)
  - Grant Sanderson (3blue1brown): Neural networks playlist
    - Beautifully animated basic math explanation (65 min)
  - Shree Nayar: https://fpcv.cs.columbia.edu/ → Perception → Neural Networks
    - Strong mathematical introduction (~1,5 h)
  - Andrew Glassner: Deep Learning Crash Course at Siggraph 2018
    - Visual introduction (no math) (~3 h)
  - Andrej Karpathy (u.a.): Stanford course from 2016
    - Practical introduction in the software development (~20 h)
  - Justin Johnson: Michigan State University course from 2020
    - Further development of the Stanford course (~24 h)
  - Alexander Amini: MIT Introduction to Deep Learning 6.S191 from 2023
    - A one week lecture including slides and coding exercises (10 h, lectures only)

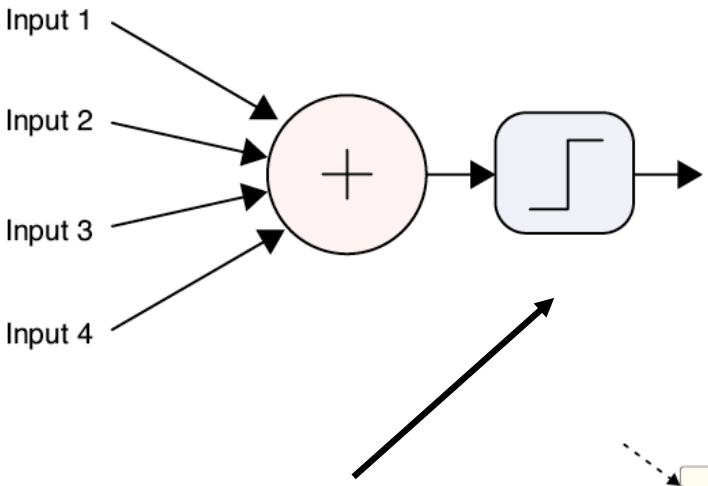
Check my page that is updated from time to time:
https://uhahne.github.io/deeplearning/

Perceptron as first idea of an artificial neuron



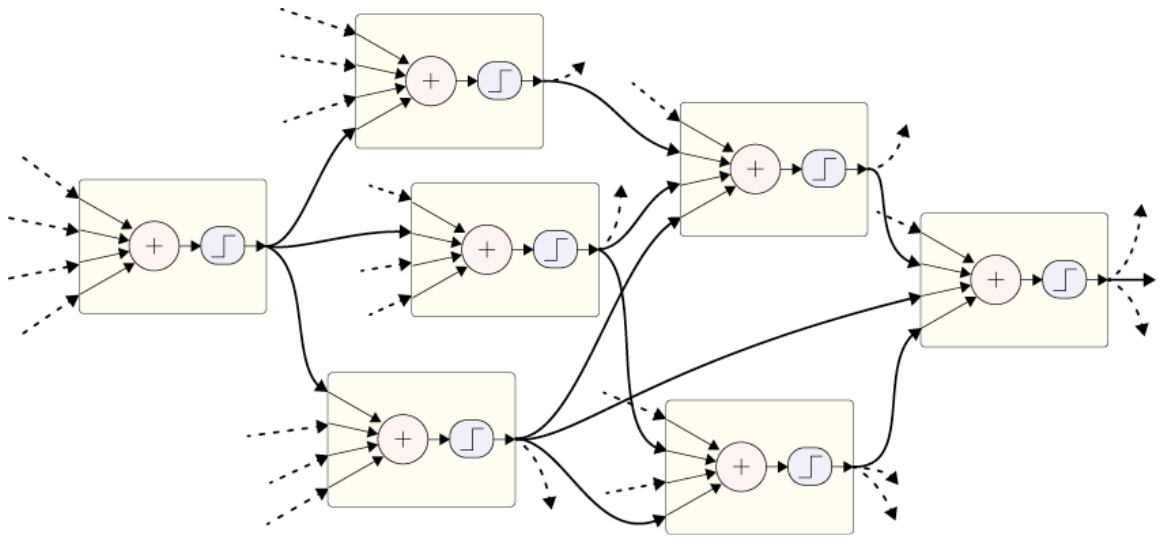
From Andrew Glassner, Deep Learning - a visual approach

Neurons



activation function

Neural network

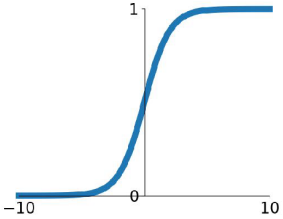


From Andrew Glassner, Deep Learning - a visual approach

Activation functions

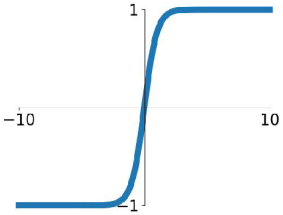
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



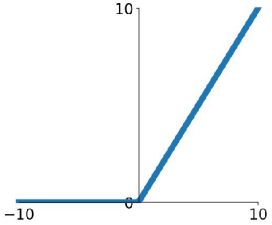
tanh

$$\tanh(x)$$



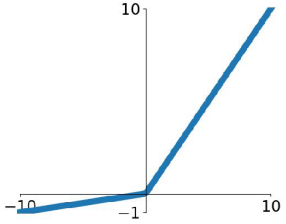
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

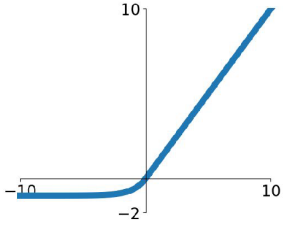


Maxout

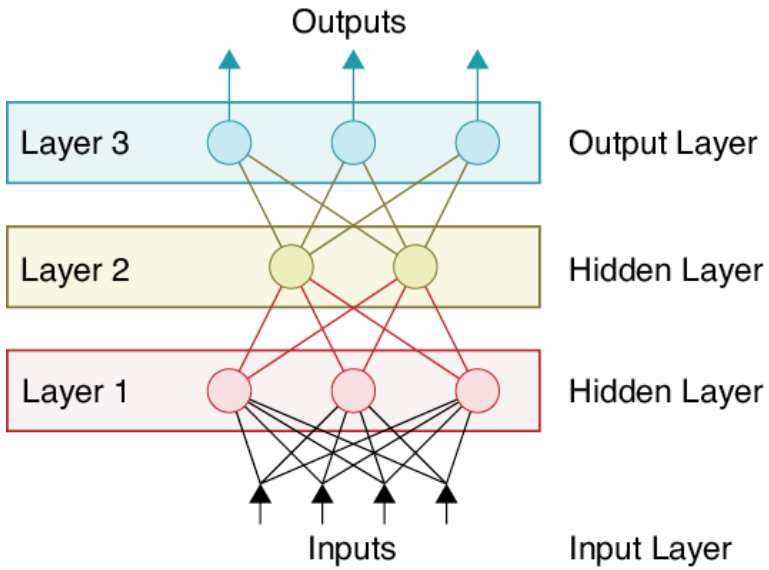
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

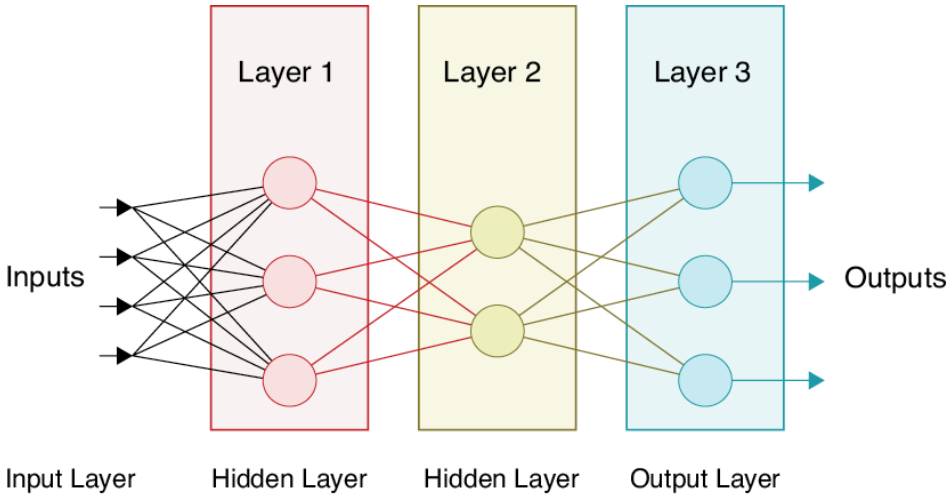


A neural network



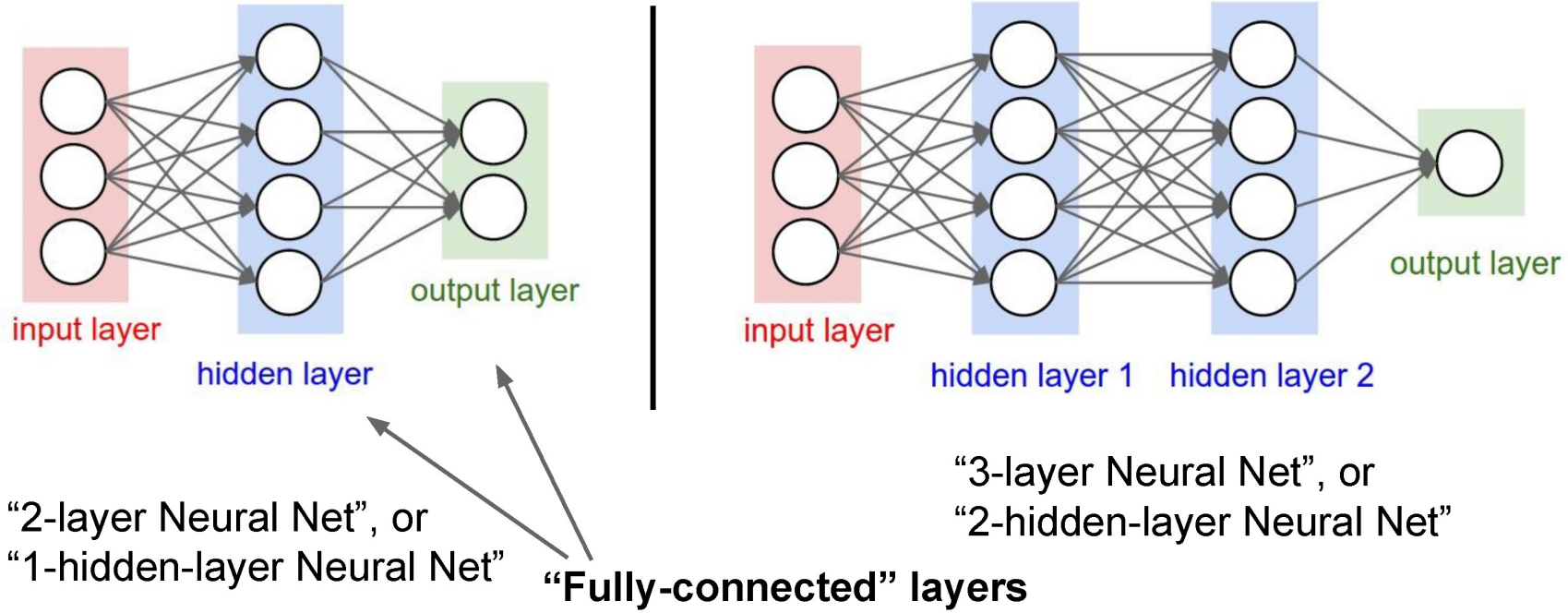
Vertical illustration

Horizontal illustration



From Andrew Glassner, Deep Learning - a visual approach

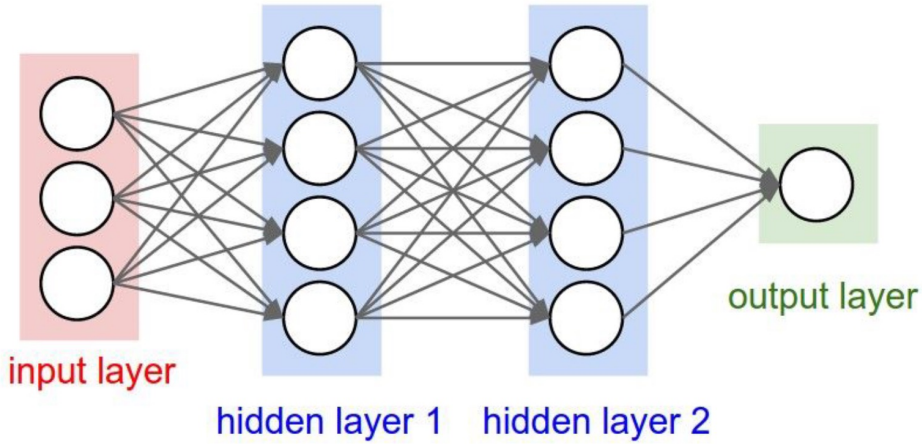
Neural network: architectures



- **Deep** neural networks typically have many layers and potentially millions of parameters.

Computation

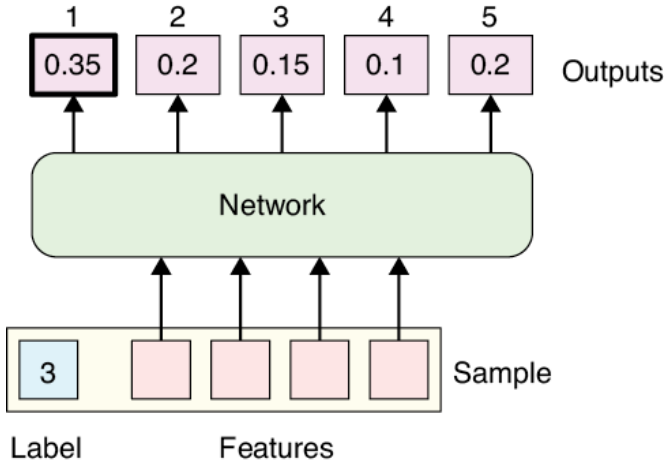
Example feed-forward computation of a neural network




```
# forward-pass of a 3-layer neural network:
f = lambda x: 1.0/(1.0 + np.exp(-x)) # activation function (use sigmoid)
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)
h1 = f(np.dot(W1, x) + b1) # calculate first hidden layer activations (4x1)
h2 = f(np.dot(W2, h1) + b2) # calculate second hidden layer activations (4x1)
out = np.dot(W3, h2) + b3 # output neuron (1x1)
```

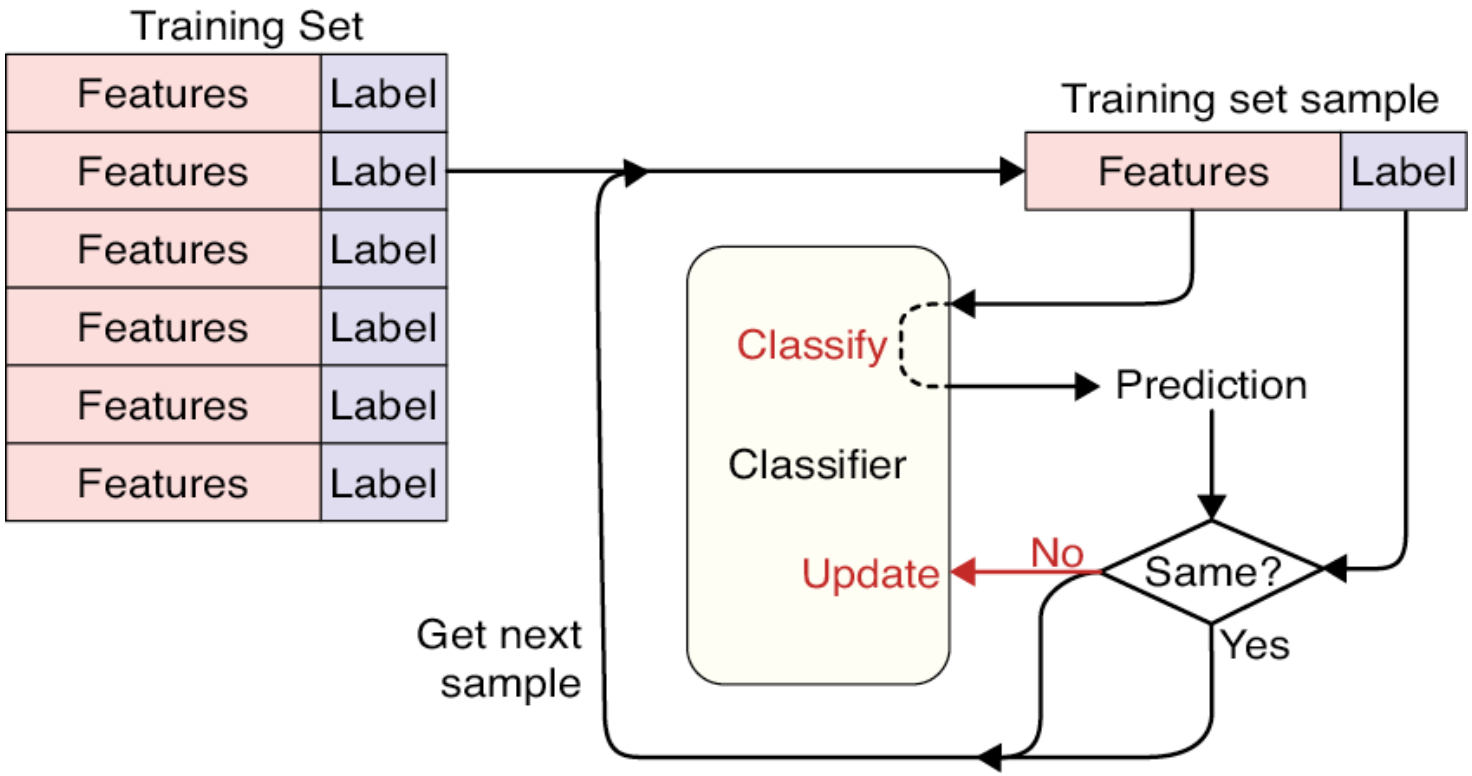
Goal of the process

- Start with random (small) values for the parameters and improve them step by step.
- **Training data**, from which the desired result is known, is fed into the network and the result is tested using **test data**.
- Networks learn by minimizing their errors. The process starts with a number called the cost or **loss** for each error. During training, the network reduces the cost, which leads to results that are closer to what one wants.



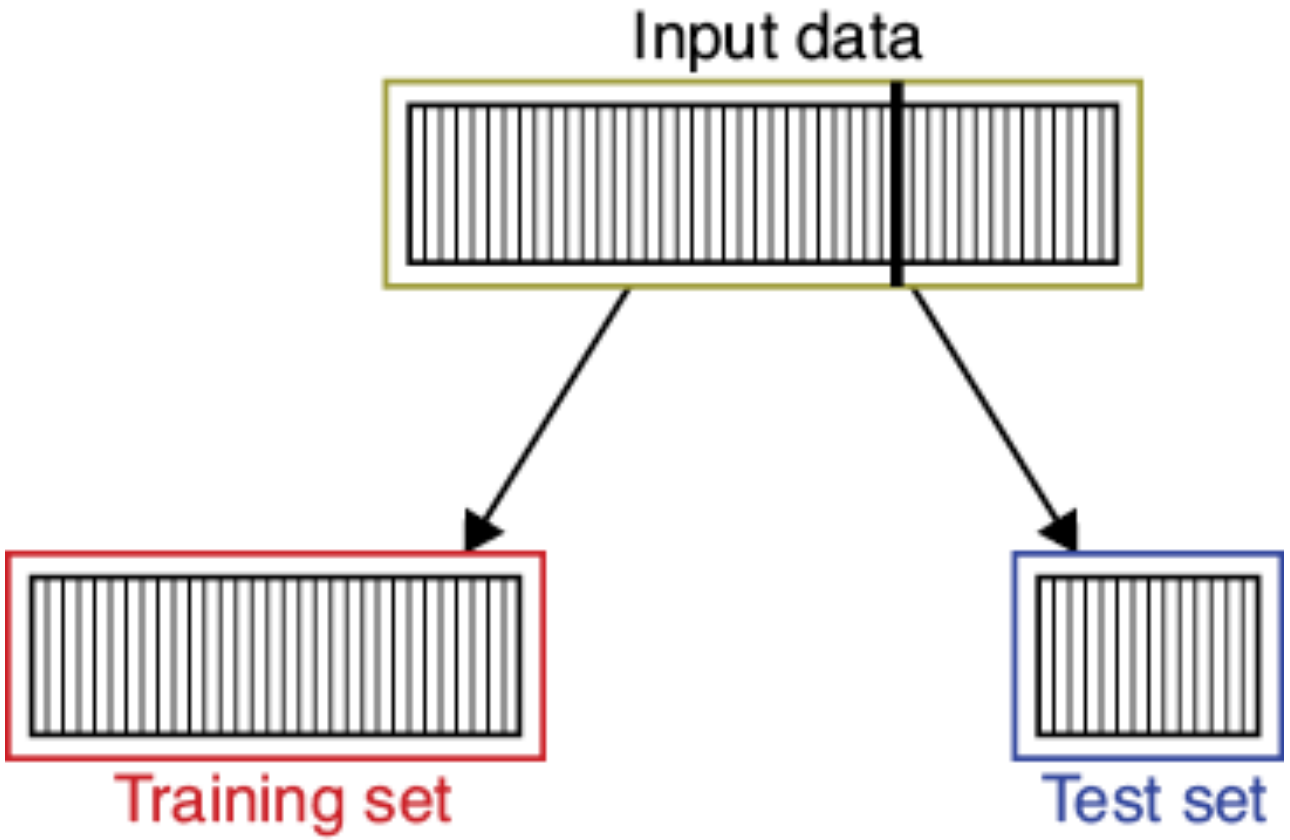
From Andrew Glassner, Deep Learning - a visual approach

Training process



From Andrew Glassner, Deep Learning - a visual approach

Separate training and test data



From Andrew Glassner, Deep Learning - a visual approach

Loss function

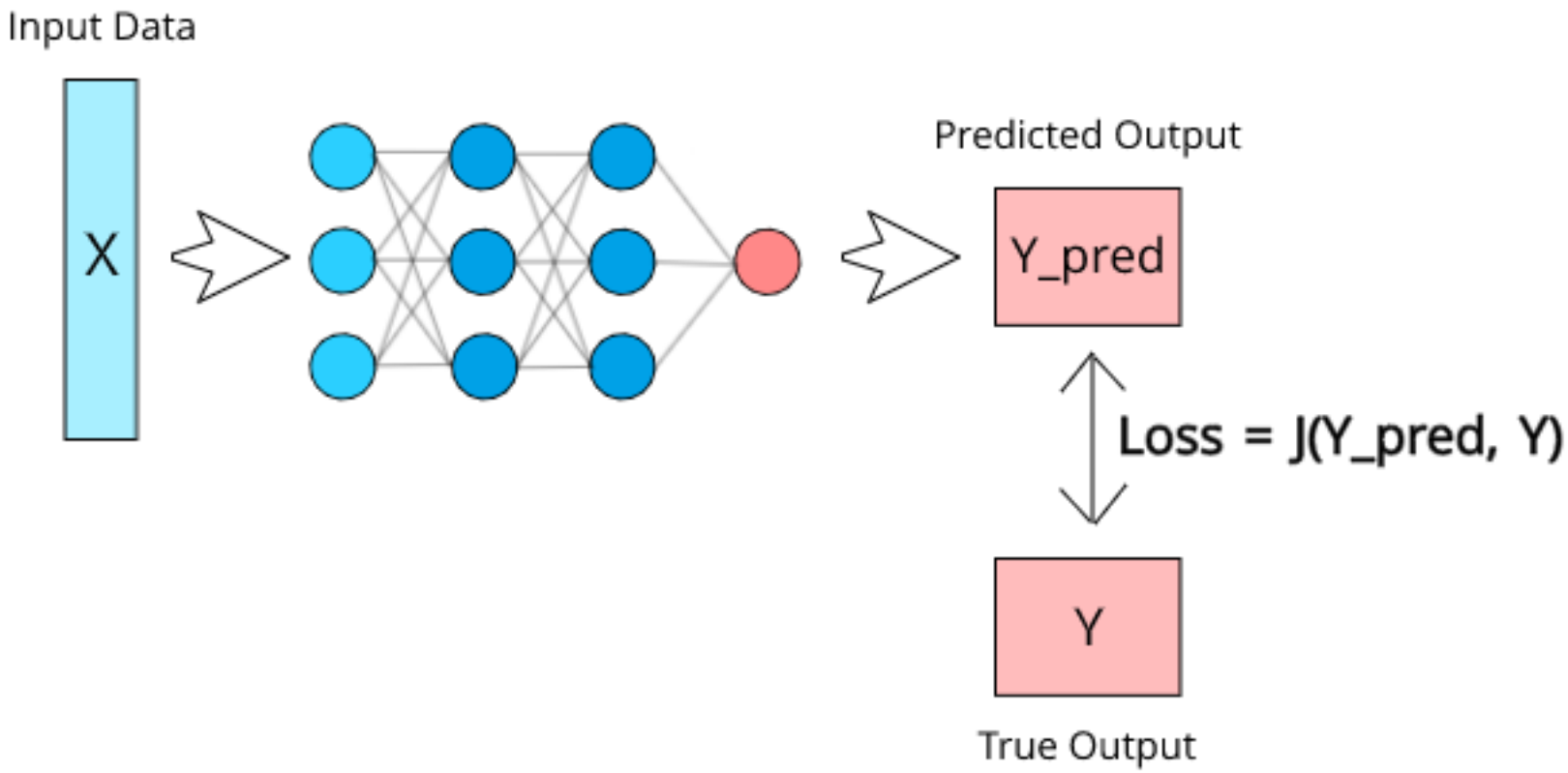
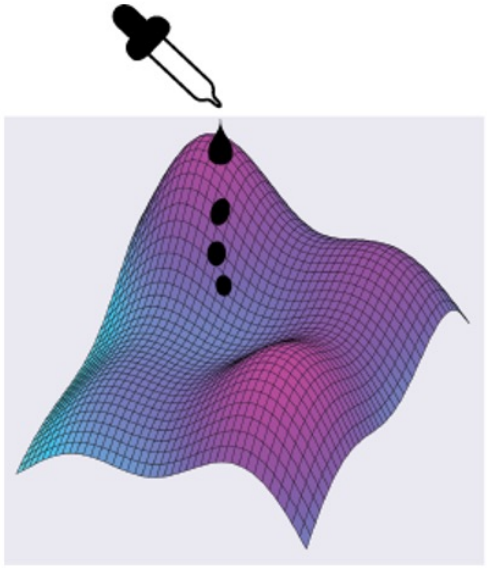
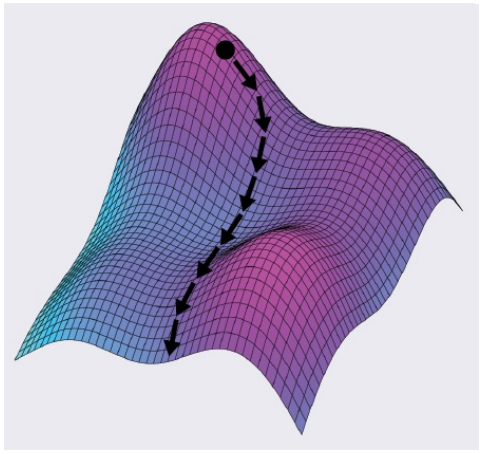


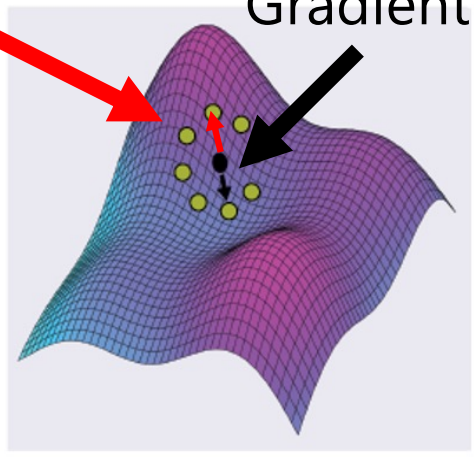
Image from the Deep Learning Bible – 2
https://wikidocs.net/book/7972

Gradient Descent

- In each step the parameters are updated.
- The parameters are changed against the direction of the steepest change. This method is called gradient descent.
- Several training steps with a lot of training data are necessary.

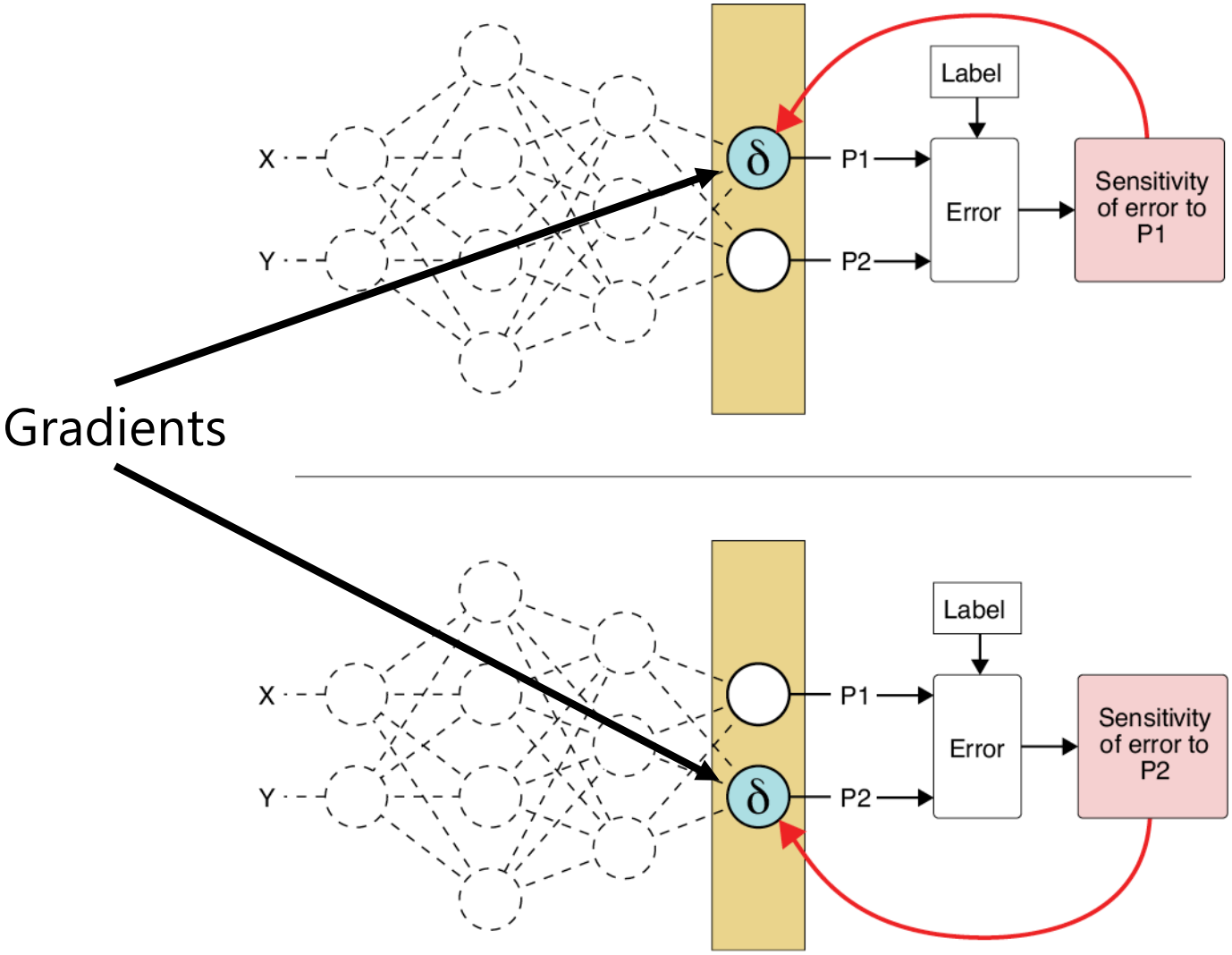






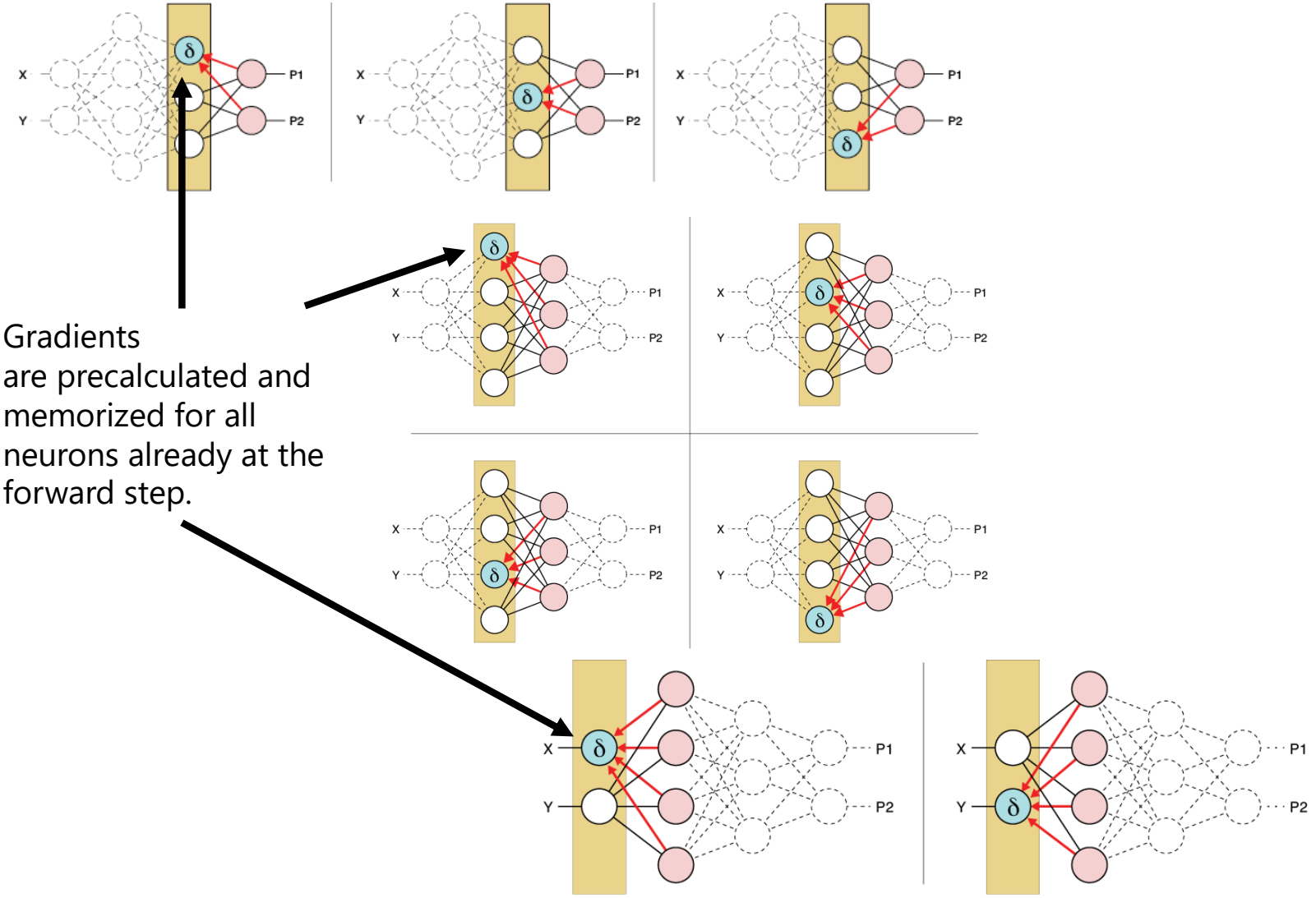
From Andrew Glassner, Deep Learning - a visual approach

Backpropagation



From Andrew Glassner, Deep Learning - a visual approach

Backpropagation



Gradients
are precalculated and
memorized for all
neurons already at the
forward step.

From Andrew Glassner, Deep Learning - a visual approach

Important terms

- Loss function
- Gradient Descent, Backpropagation (Backprop)
  - Local minima are not a practical problem for large networks.
- Activation functions
  - Sigmoid (only interesting for historical reasons)
  - tanh (no longer really relevant in practice)
  - ReLU (default)
- Layers as building blocks of the networks
  - TensorFlow offers over 250 different layer classes: https://www.tensorflow.org/api_docs/python/tf/keras/layers
  - PyTorch similarly many: https://pytorch.org/docs/stable/nn.html
- Hyperparamters: batch, epoch, learning rate (mostly η [eta]).
- more at https://developers.google.com/machine-learning/glossary

Depth using deep learning

- Basic idea is using supervised learning with many highly variant images
  - Large data sets + augmentation

- Evolution in last 10 years:
  - [Eigen2014] – first paper with promising results
  - [Ranftl2022] – MiDaS v3 → much better results
  - [Yang2024] – DepthAnything → incredibly good results

Data sets

A large number of color images together with depth maps.

In a depth map, each pixel has a distance value instead of a color value.

Depth maps can be absolute (real metric values) or relative (closest point 0, farthest point 1 or vice versa)

- List of 230 RGB + Depth data sets
- Popular ones:
  - NYU Depth Dataset V2, 2012, 407K images
  - Kitti, 2018, 93K images

Augmentation

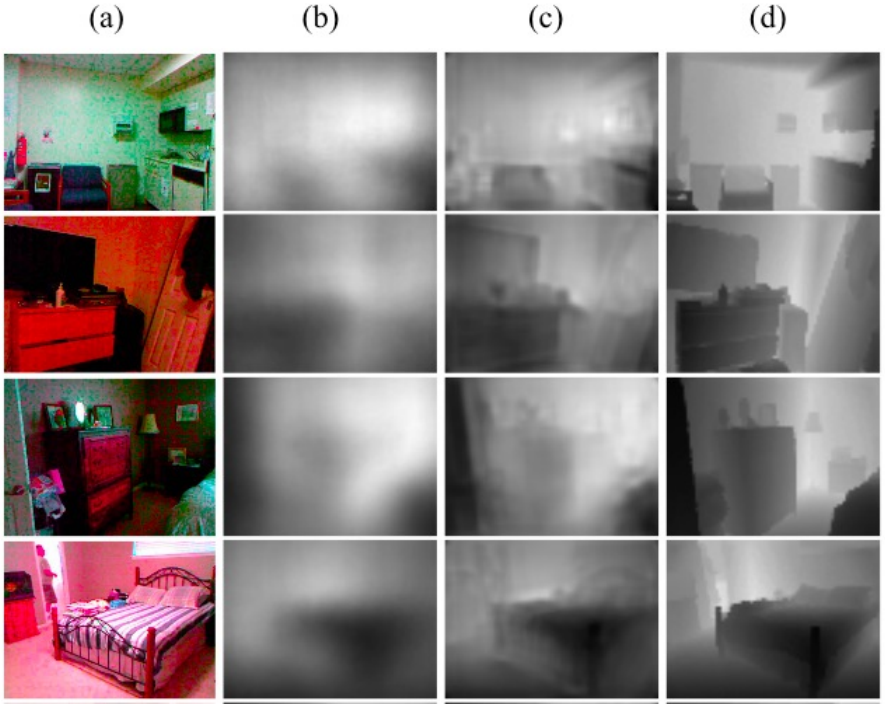
Data augmentation is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data.

Augmented vs. Synthetic data

- **Augmented data** is driven from original data with some minor changes. In the case of image augmentation, we make geometric and color space transformations (flipping, resizing, cropping, brightness, contrast) to increase the size and diversity of the training set.
- **Synthetic data** is generated artificially without using the original dataset.

(source: https://www.datacamp.com/tutorial/complete-guide-data-augmentation)

Eigen2014



- Using NYU Depth V2 and Kitti
- Tested on these data sets only.

(a) Input
(b) Coarse result
(c) Fine result
(d) Ground truth

Ranftl2022 (MiDaS)

- https://www.youtube.com/watch?v=D46FzVyL9I8

- **Zero-shot cross-dataset transfer:** train a model on certain datasets and then test its performance on other datasets that were never seen during training.
- approximately six GPU months of computation

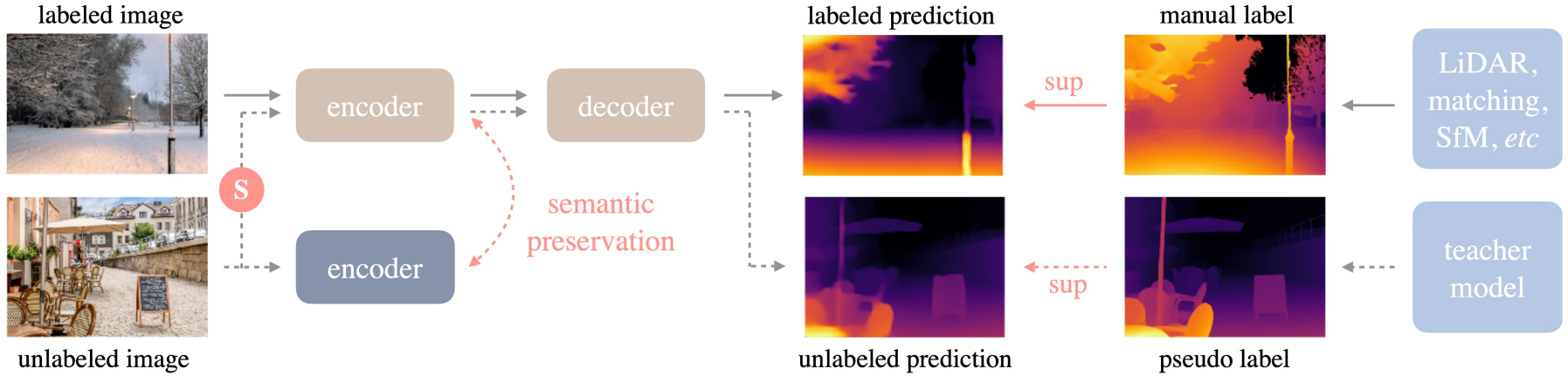
Yang2024 (DepthAnything)

- Done at TikTok.
- They used even more data sets and a way to include unlabelled images.
- Trained on a combination of 1.5M labeled images and **62M+ unlabeled images**.

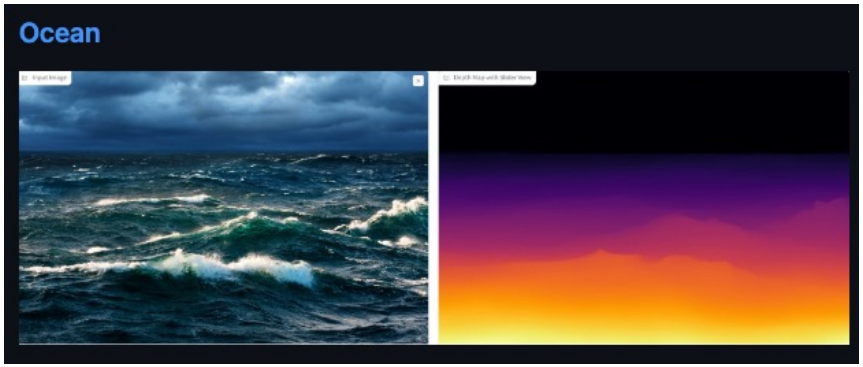
Comparison between Depth Anything and MiDaS v3.1

Yang2024 (DepthAnything)

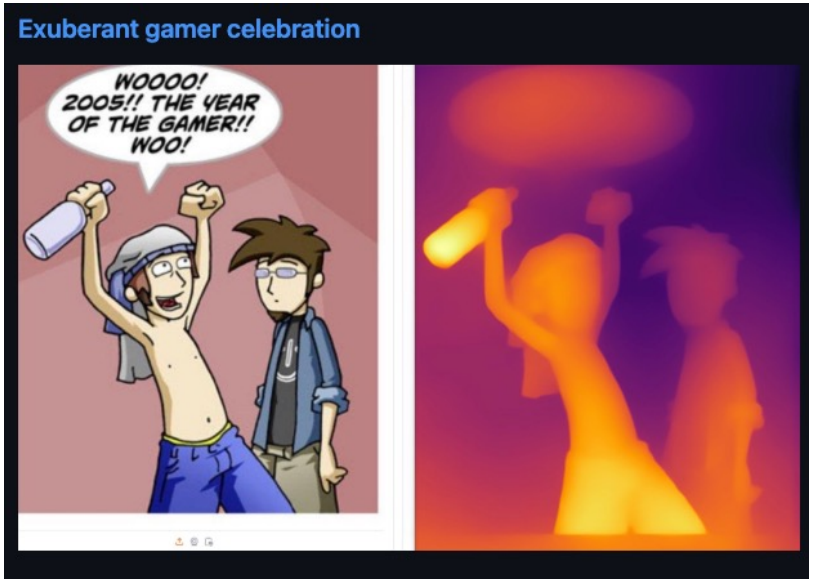
- Code available: https://github.com/LiheYoung/Depth-Anything?tab=readme-ov-file
- Learn to use it: https://learnopencv.com/depth-anything/



Yang2024 (DepthAnything)



- It also works for cartoons and (AI) generated content
- Gallery with more results



Yang2024v2 (Depth Anything V2)

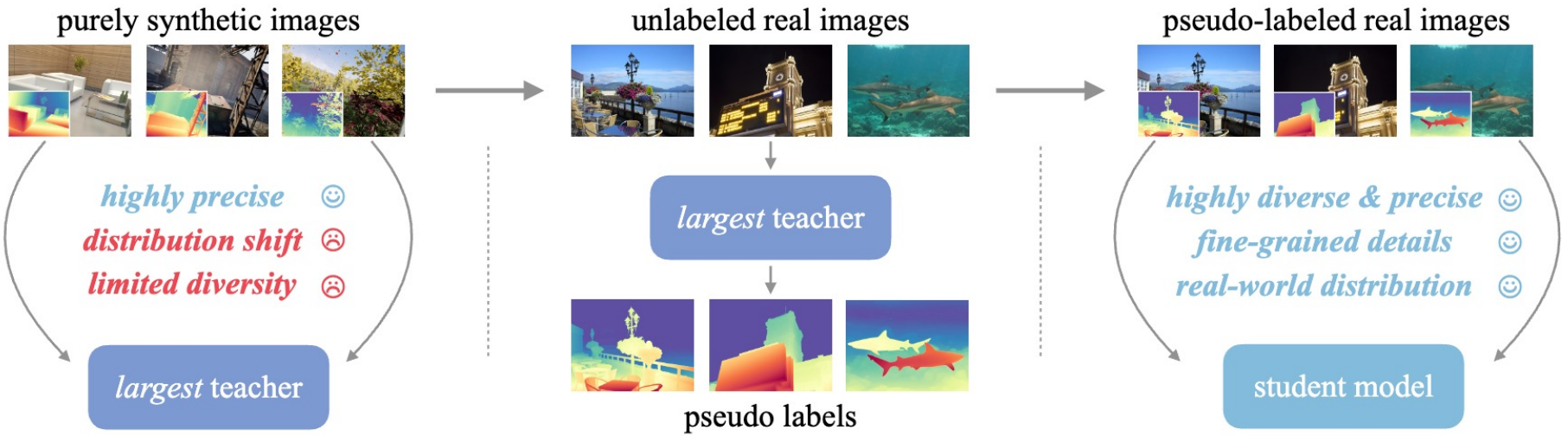


Figure 7: Depth Anything V2. We first train the most capable teacher on precise synthetic images. Then, to mitigate the distribution shift and limited diversity of synthetic data, we annotate unlabeled real images with the teacher. Finally, we train student models on high-quality pseudo-labeled images.

- Even more images for training

Bibliography

- [Eigen2014]
  - Eigen, D., Puhrsch, C., & Fergus, R. (2014). *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network* (arXiv:1406.2283). arXiv. https://doi.org/10.48550/arXiv.1406.2283
- [Ranftl2022]
  - Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2022). Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(3), 1623–1637. https://doi.org/10.1109/TPAMI.2020.3019967
- [Yang2024]
  - Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., & Zhao, H. (2024). *Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data* (arXiv:2401.10891). arXiv. https://doi.org/10.48550/arXiv.2401.10891
- [Yang2024v2]
  - Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). *Depth Anything V2*, NeurIPS 2024, https://arxiv.org/abs/2406.09414