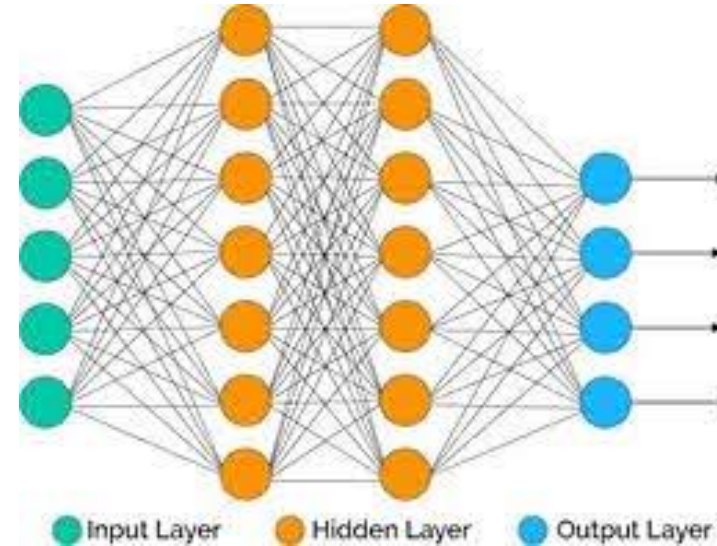












おきすつなはまやれを

LAB 4: IMAGE CLASSIFICATION WITH NEURAL NETWORKS

MACHINE LEARNING 2023-24
(P. ZANUTTIGH – ICT+PHYSICS OF DATA)

IMAGE CLASSIFICATION WITH NEURAL NETWORKS



0	T-shirt	
1	Trouser	
2	Pullover	
3	Press	
4	coat	
5	sandal	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boot	

Task to be solved:

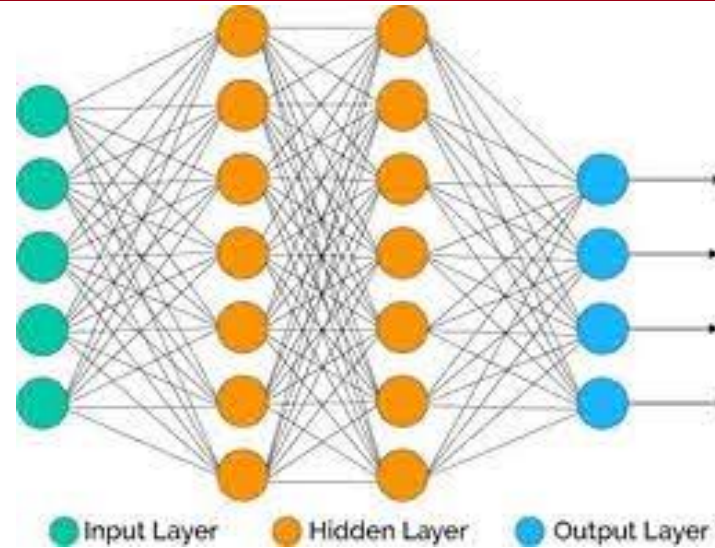
Classification of Kuzushiji (old Japanese) characters using Neural Networks

THE KMNIST DATASET

- Small images of Japanese characters
- 70'000 samples (7'000 for each class)
- Each sample is a 28x28 grayscale image (784 pixels, i.e., $x \in \mathbb{R}^{784}$)
- Divided into 60'000 for training and 10'000 for testing
- Classified into 10 classes corresponding to 10 different characters
- Expect an accuracy around 70-80% for a «baseline» NN classification



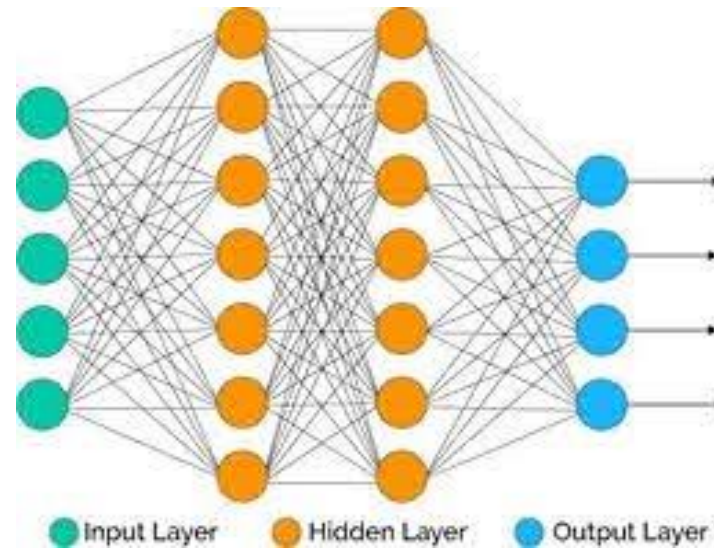
CLASSIFICATION OF CHARACTERS



おきすつなはまやれを

- Dataset of small pictures of characters: multi-class classification
- Start by using Neural Networks in scikit-learn
- Try different network architectures (e.g., change number of neurons and layers)
- Not always the largest is the best, specially if training data is limited
- See the impact of the learning rate
- **Learning rate**: controls speed of convergence vs stability trade-off

FORWARD AND BACKWARD PROPAGATION



おきすつなはまやれを

- The notebook contains the implementation of forward and backward propagation
- Implement two simple activation functions (ReLU and sigmoid)
- Implement the key steps of the forward propagation (linear computation in the neurons, then use the previously implemented activations)
 - *Note-1-: the proposed implementation has an explicit bias (instead of homogeneous model as in the theory)*
 - *Note-2-: be careful of the required data structures in input/output (see the comments in the code)*
- The more complex backward propagation is already implemented
- Then train and use your “custom” neural network!

- Complete the jupyter notebook
 - FIRST THING TO DO: you need to put your name and ID number in the notebook
 - You can use the ID also as seed for random number generators, try different seeds
 - The notebook has **missing code**: need to fill in what is missing
 - You must write the **answer to all the questions** in the notebook
 - But do not change the structure or the input data files, they will not be submitted
- Check that the notebook run properly from the beginning with the provided data
 - **use the "restart kernel&run all" command**
- Save them as **surname_name_lab4.ipynb**
- Submit on elearning
- **Lab 4 : 22/12 h12,30** in rooms Te,Ue
- Delivery deadline: 10/1

