**CNN**

* **Yêu cầu**

**+ Hiển thị các feature map của từng lớp embedding**

**+ Hiển thị kích thước đầu vào + đầu ra của một lớp bất kì**

**+ Custom hàm loss, optimazer theo yêu cầu**

**+ So sánh kết quả giữa các optimazer khác nhau**

**+ Xuất ra trọng số epoch bất kì**

**+ Yêu cầu traning với epoch đề cho**

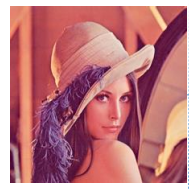
***+ Chú ý câu tính param từ kiến trúc của mô hình (RNN - LSTM, CNN)***

**BÀI TẬP:**

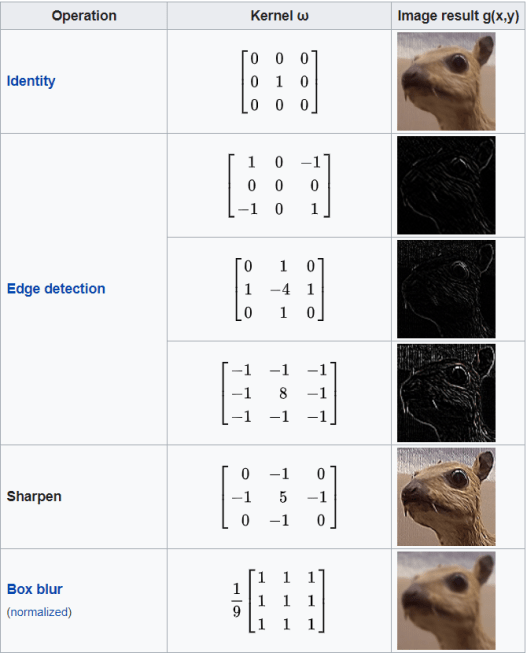
**Tham khảo :** [**https://lme.tf.fau.de/lecture-notes/lecture-notes-dl/lecture-notes-in-deep-learning-loss-and-optimization-part-3/**](https://lme.tf.fau.de/lecture-notes/lecture-notes-dl/lecture-notes-in-deep-learning-loss-and-optimization-part-3/)

Xây dựng mô hình huấn luyện với các yêu cầu sau:

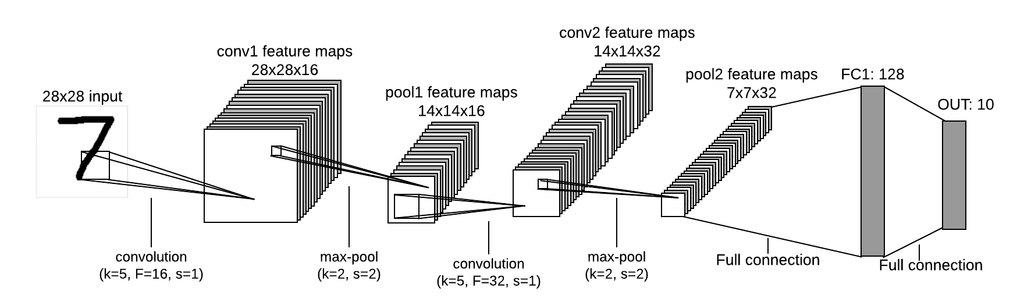
1. Input:



2. Chọn 1 Kernel trong danh sách sau:



3. Max pooling (2x2)



Mô tả kiến trúc :

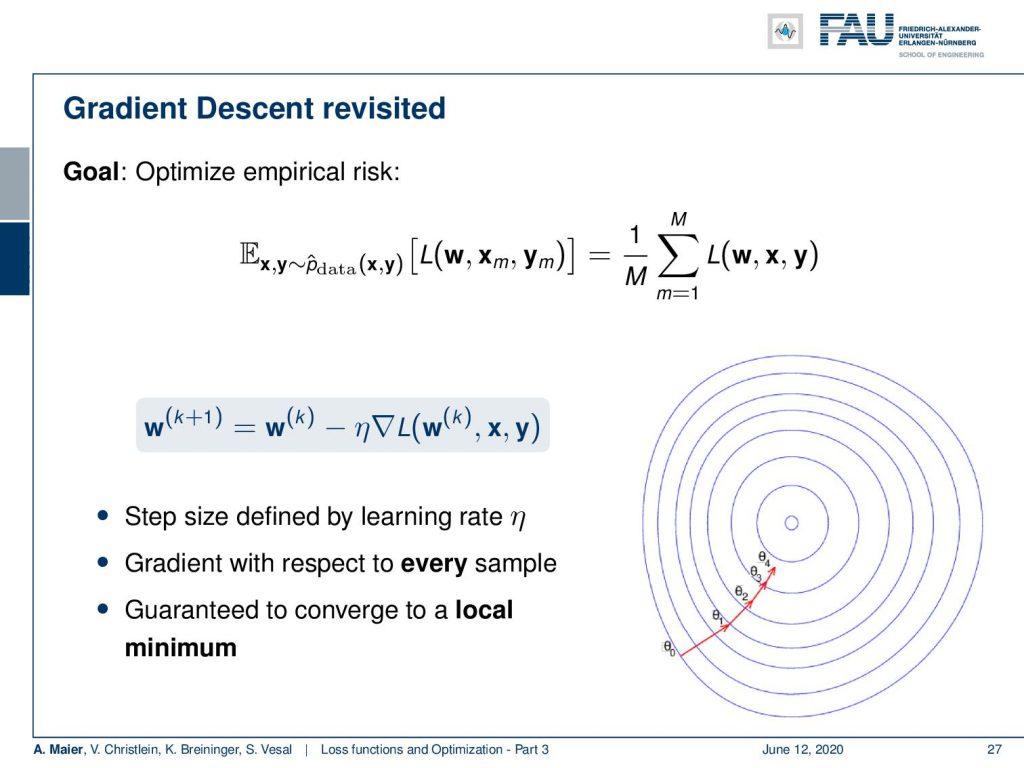
* Input đầu vào là ma trận ảnh 28\*28\*3
* Conv (k=3,F=16,s=1) thu được 28\*28\*16
* Max-pool : (k=2,s=2) thu được 14\*14\*16
* Conv (k=3,F=32,s=1) thu được 14\*14\*32
* Max-pool (k=2,s=2) thu được 7\*7\*32
* Fullconected – Flatent thu được : 128
* Output : 10 label

Hô hình CNN đề xuất

Yêu cầu bài toán:

1. Hiển thị map các feature map
2. Hiển thị kích thước vector flatting
3. Lớp full connection sử dụng các hàm: Loss loss function, hàm optimazer:

Gradienet descent revisited



1. So sánh kết quả trên khi sử dụng adam hoặc sgd.
2. Hiển thị weight sau khi mô hình hội tụ