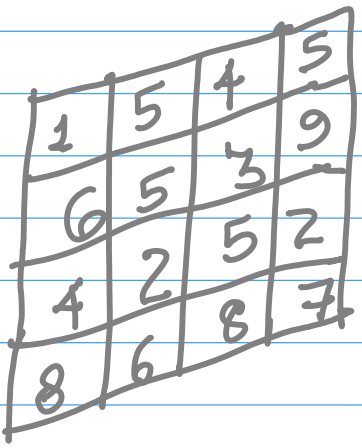
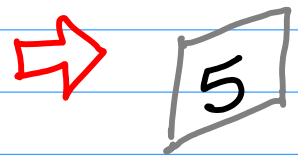
5) Global Average Pooling



The entire frame into one

$\Sigma = 80$, $80/4 \times 4 = 5$



(16 pixels: 4x4 grid)

why Pooling? — contraction
— Invariance. Invariance to small transformations, distortions, translation. A small distortion in the input will not change the outcome of pooling drastically since we take the max / average value in a local neighbourhood

LOCAL RESPONSE NORMALISATION (LRN)

from neuro-biology (AlexNet)

→ 'lateral inhibition' → an excited neuron inhibits its neighbours → creates contrast in the area and increases sensory perception

→ No solid mathematical background

→ motivated by biology (evolution), shown statistically.

Batch Normalisation (2015) ["BN"]

[Ioffe & Szegedy, 2015]

→ Addresses the problem of **Internal Covariate Shift** (initially believed) current opinion: works because it smooths the objective function.

At initialisation: actually induces severe gradient explosion, which is only alleviated by skip connections in residual networks

The basic issue (historically) Each layer of a neural network has inputs with a corresponding distribution. This is affected during the training process by the randomisation

- the parameter initialisation
- the input

- During training, as parameters of the preceding layers change, the distribution of inputs to the current layer changes accordingly
- The current layer needs to constantly adjust to new distributions.
- small changes in the initial layers amplify, and result in a significant shift in deeper hidden layers

BENEFITS:

- Reduce unwanted shifts to speed up training
- permits a higher learning rate without vanishing/exploding gradients
- Regularisation effect: unneces ary to use 'dropout' to mitigate overfitting
- Robustness to different initialisation schemes and learning rates

Stochastic Optimisation → not all training data is taken together → normalisation is done in batches. Batch B has size m

$$\mu_B = \frac{1}{m}\sum_{n=1}^{m} x^{(n)} \; ; \; \sigma_B^2 = \frac{1}{m}\sum_{n=1}^{m}\left[x^{(n)} - \mu_B\right]^2$$

x: x is a scalar → one of the input dimensions

call it x_i, say.

$$\hat{x}_i \triangleq \frac{x_i - \mu_{B_i}}{\sqrt{\sigma_{B_i}^2 + \epsilon}}$$

ϵ → small positive constant used for numerical stability

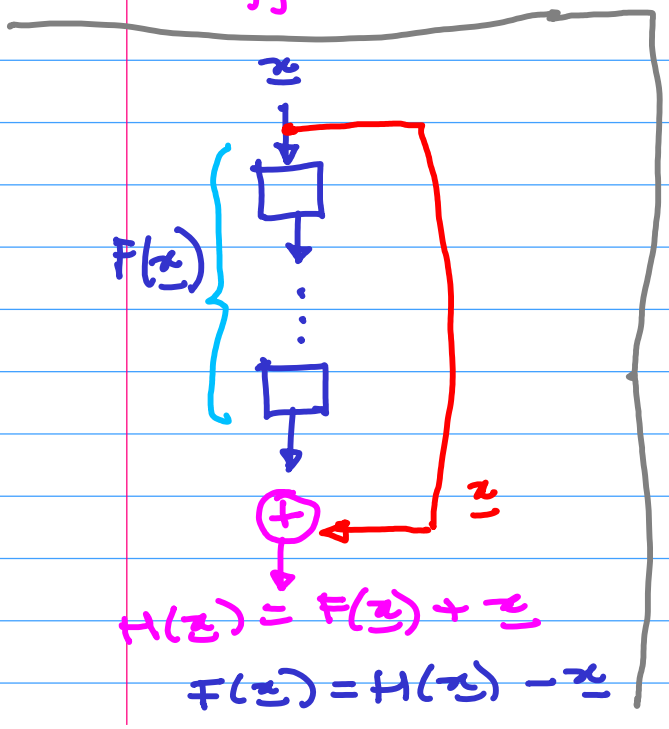
$$y_i \triangleq \gamma_i \hat{x}_i + \beta_i$$

γ_i, β_i → trainable parameters

Residual Connections / Skip Connections / Highway networks

Trend towards deeper neural networks → better accuracy or performance

trade-off: harder for training to converge



Used widely →

ResNet (Computer Vision),
Transformer (NLP, Computer vision)
AlphaZero (RL),
AlphaFold (Protein Structure prediction)

Issues:

- Difficult to learn an identity mapping across layers
- Training a deep network is difficult because of exploding and vanishing gradients
- Observation: convolutional layers are often better at learning the residual rather than learning the feature map directly.

(*) Residual connections / skip connections / highway connections (contd.)

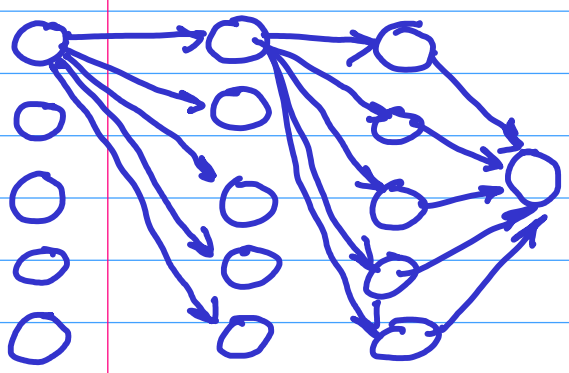
(*) The magnitude of the problem: →

- AlexNet (2012) had 5 convolutional layers
- 2014 : VGG, GoogleNet
  - VGG → 19 layers
  - GoogleNet → 22 layers
- ResNet (34 → 50 layers deep architectures) deeper but had overall lower complexity.

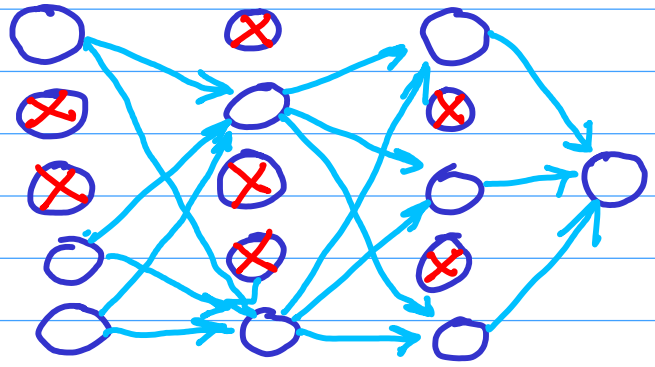
(*) "DROPOUT" (AlexNet 2012)

Heuristic : applied at the training phase (FC layers)

Dropout is a kind of regularisation technique to reduce overfitting.



Usual FC scenario



Scenario with dropout

AlexNet : $p = 0.5$ at the first two fully connected layers

Neuron : has a probability not to contribute to the feedforward phase & participate in the backpropagation. ⇒ Each neuron can have a larger chance to be trained, and not depend on some 'strong' neuron. No dropout at the test time