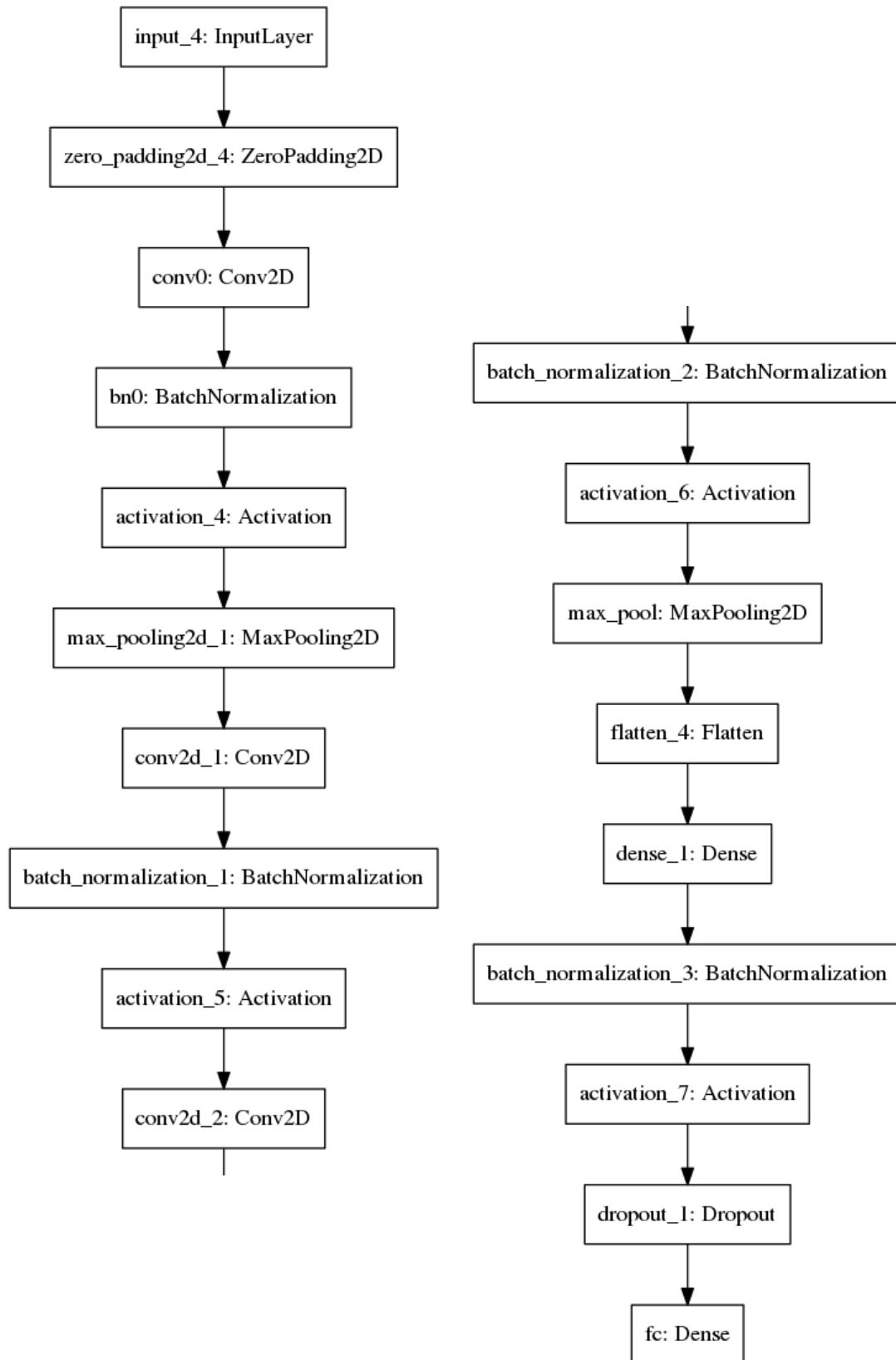


Convolutional neural networks – week 2

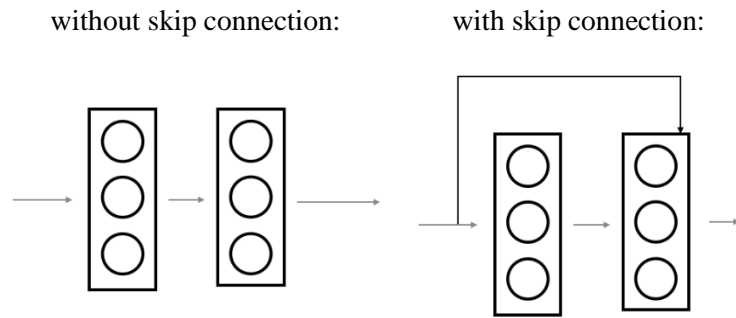
Emotion Tracking – CNN model implemented using keras

Classify emotions (smiling or not smiling) based on RGB pictures: $y = \begin{cases} 0 & \text{not smiling} \\ 1 & \text{smiling} \end{cases}$



Residual Networks

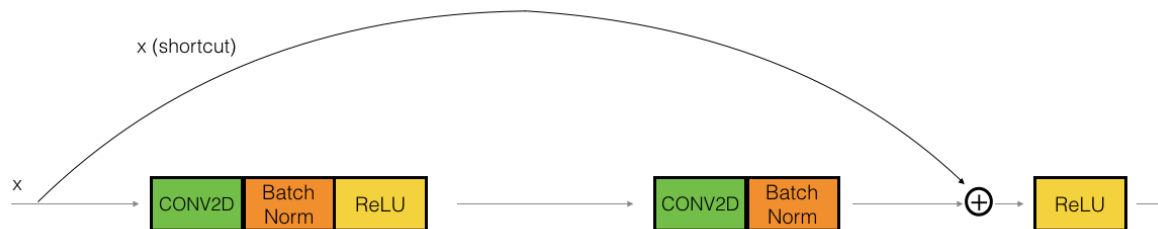
"skip connection" allows the model to skip layers:



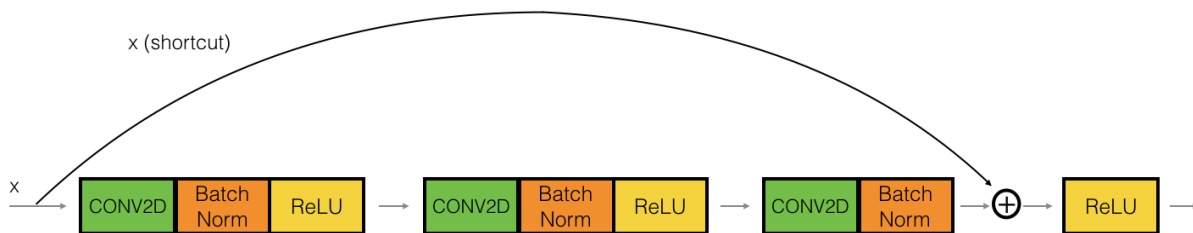
The identity block

The identity block is the standard block used in ResNets, and corresponds to the case where the input activation ($a^{[l]}$) has the same dimension as the output activation ($a^{[l+1]}$). BatchNorm step is added to speed up training.

“skip” over 2 layers:



“skip” over 3 layers:



The individual steps – implementation of `identity_block()`:

First component of main path:

- The first CONV2D has F_1 filters of shape (1,1) and a stride of (1,1). Its padding is "valid" and its name should be `conv_name_base + '2a'`. Use 0 as the seed for the random initialization.
- The first BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '2a'`.
- Then apply the ReLU activation function. This has no name and no hyperparameters.

Second component of main path:

- The second CONV2D has F_2 filters of shape (f, f) and a stride of (1,1). Its padding is "same" and its name should be `conv_name_base + '2b'`. Use 0 as the seed for the random initialization.
- The second BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '2b'`.
- Then apply the ReLU activation function. This has no name and no hyperparameters.

Third component of main path:

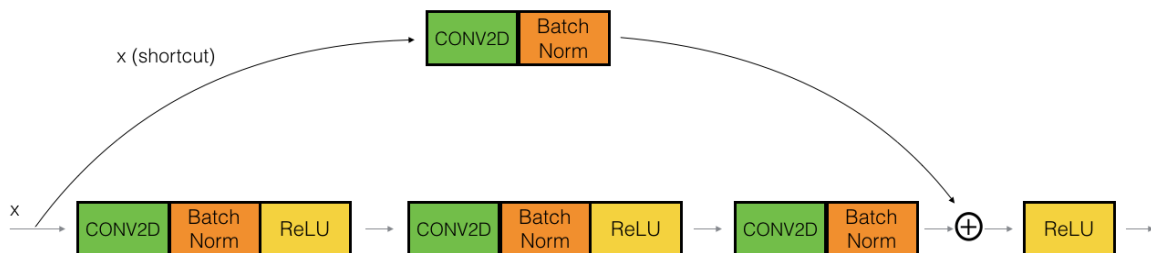
- The third CONV2D has F_3 filters of shape (1,1) and a stride of (1,1). Its padding is "valid" and its name should be `conv_name_base + '2c'`. Use 0 as the seed for the random initialization.
- The third BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '2c'`.
- **Note** – There is **no** ReLU activation function in this component.

Final step:

- The `X_shortcut` and the output from the 3rd layer `X` are added together.
(The syntax looks something like `Add()([var1, var2])`)
- Apply the ReLU activation function. This has no name and no hyperparameters.

The convolutional block

Use this type of block when the input and output dimensions don't match up:



The CONV2D layer in the shortcut path is used to resize the input `x` to a different dimension, so that the dimensions match up in the final addition needed to add the shortcut value back to the main path.

The CONV2D layer on the shortcut path does not use any non-linear activation function. Its main role is to just apply a (learned) linear function that reduces the dimension of the input, so that the dimensions match up for the later addition step.

The details of the convolutional block – implementation of `convolutional_block()`:

First component of main path:

- The first CONV2D has F_1 filters of shape (1,1) and a stride of (s,s). Its padding is "valid" and its name should be `conv_name_base + '2a'`. Use 0 as the `glorot_uniform` seed.
- The first BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '2a'`.
- Apply ReLU activation function. This has no name and no hyperparameters.

Second component of main path:

- The second CONV2D has F_2 filters of shape (f,f) and a stride of (1,1). Its padding is "same" and its name should be `conv_name_base + '2b'`. Use 0 as the `glorot_uniform` seed.
- The second BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '2b'`.
- Apply ReLU activation function. This has no name and no hyperparameters.

Third component of main path:

- The third CONV2D has F_3 filters of shape (1,1) and a stride of (1,1). Its padding is "valid" and its name should be `conv_name_base + '2c'`. Use 0 as the `glorot_uniform` seed.
- The third BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '2c'`.
- **Note** – There is **no** ReLU activation function in this component.

Shortcut path:

- The CONV2D has F_3 filters of shape (1,1) and a stride of (s,s). Its padding is "valid" and its name should be `conv_name_base + '1'`. Use 0 as the `glorot_uniform` seed.
- The BatchNorm is normalizing the 'channels' axis. Its name should be `bn_name_base + '1'`.

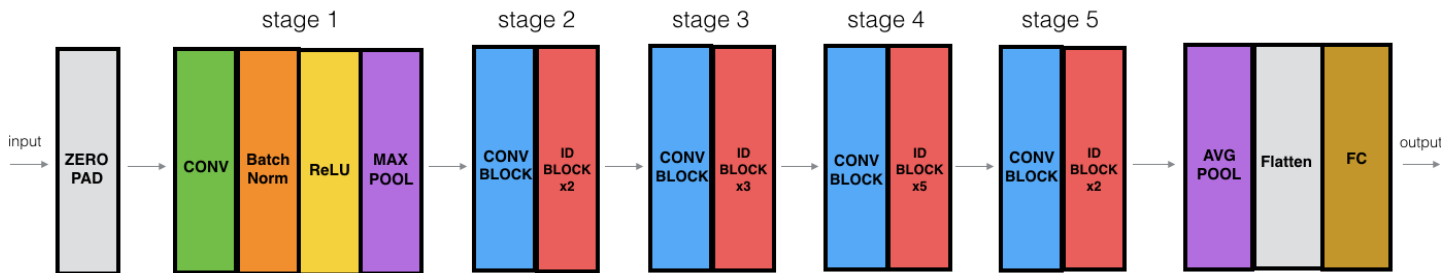
Final step:

- The shortcut and the main path values are added together.
- Apply ReLU activation function. This has no name and no hyperparameters.

ResNet model (50 layers) – ResNet50

The architecture of the neural network:

- ! "ID BLOCK" in the diagram stands for "Identity block", "ID BLOCK x3" means you should stack 3 identity blocks together.



The details of the model:

Zero-padding pads the input with a pad of (3,3)

Stage 1:

- The 2D Convolution has 64 filters of shape (7,7) and uses a stride of (2,2). Its name is "conv1".
- BatchNorm is applied to the 'channels' axis of the input.
- MaxPooling uses a (3,3) window and a (2,2) stride.

Stage 2:

- The convolutional block uses three sets of filters of size [64,64,256], "f" is 3, "s" is 1 and the block is "a".
- The 2 identity blocks use three sets of filters of size [64,64,256], "f" is 3 and the blocks are "b" and "c".

Stage 3:

- The convolutional block uses three sets of filters of size [128,128,512], "f" is 3, "s" is 2 and the block is "a".
- The 3 identity blocks use three sets of filters of size [128,128,512], "f" is 3 and the blocks are "b", "c" and "d".

Stage 4:

- The convolutional block uses three sets of filters of size [256, 256, 1024], "f" is 3, "s" is 2 and the block is "a".
- The 5 identity blocks use three sets of filters of size [256, 256, 1024], "f" is 3 and the blocks are "b", "c", "d", "e" and "f".

Stage 5:

- The convolutional block uses three sets of filters of size [512, 512, 2048], "f" is 3, "s" is 2 and the block is "a".
- The 2 identity blocks use three sets of filters of size [512, 512, 2048], "f" is 3 and the blocks are "b" and "c".

The 2D Average Pooling uses a window of shape (2,2) and its name is "avg_pool".

The 'flatten' layer doesn't have any hyperparameters or name.

The Fully Connected (Dense) layer reduces its input to the number of classes using a softmax activation. Its name should be 'fc' + str(classes).