Deep NN

05 January 2023 09:16

$$\text{new wt} = \text{old wt} - \eta \frac{\partial J}{\partial w}$$

Input → L_1 → L_2 → - - - - L_{50} → output

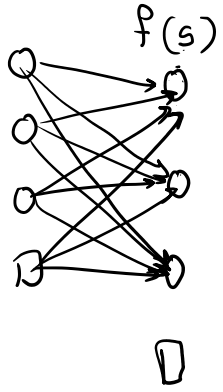
Gradient $\frac{\partial J}{\partial w}$ may decrease : $0.6 - 0.01(0.00004)$

Problem of Vanishing Gradient

Gradient may increase :

Problem of Exploding Gradient

ReLU $f(x) = \max(0, x)$



$$LeakyReLU_{\alpha}(z) = \max(\alpha z, z)$$

If $z_{\text{sum}}^{\text{wt}} = -7.8$

$\alpha = 0.01$

$Act(z) = \max((0.01)(-7.8), -7.8)$

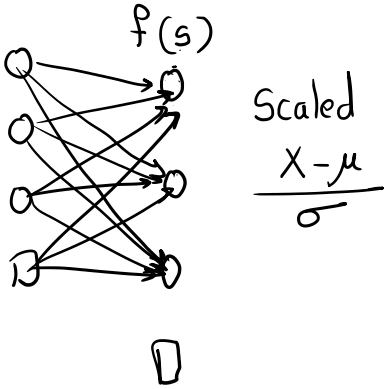
Optimizers : Method of updating the weights

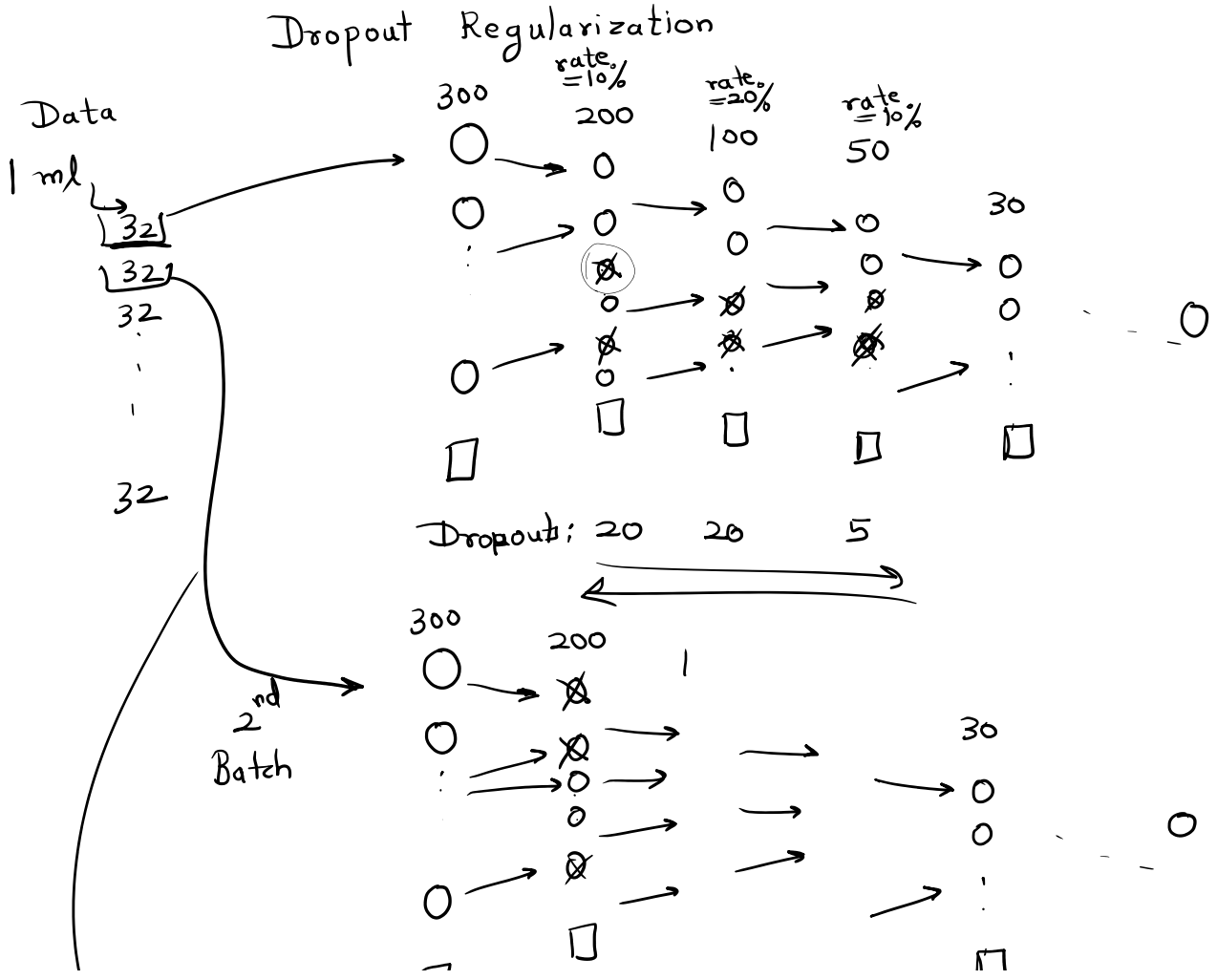
Ordinary Gradient Descent : $\text{new wt} := \text{old wt} - \eta \frac{\partial J}{\partial w}$

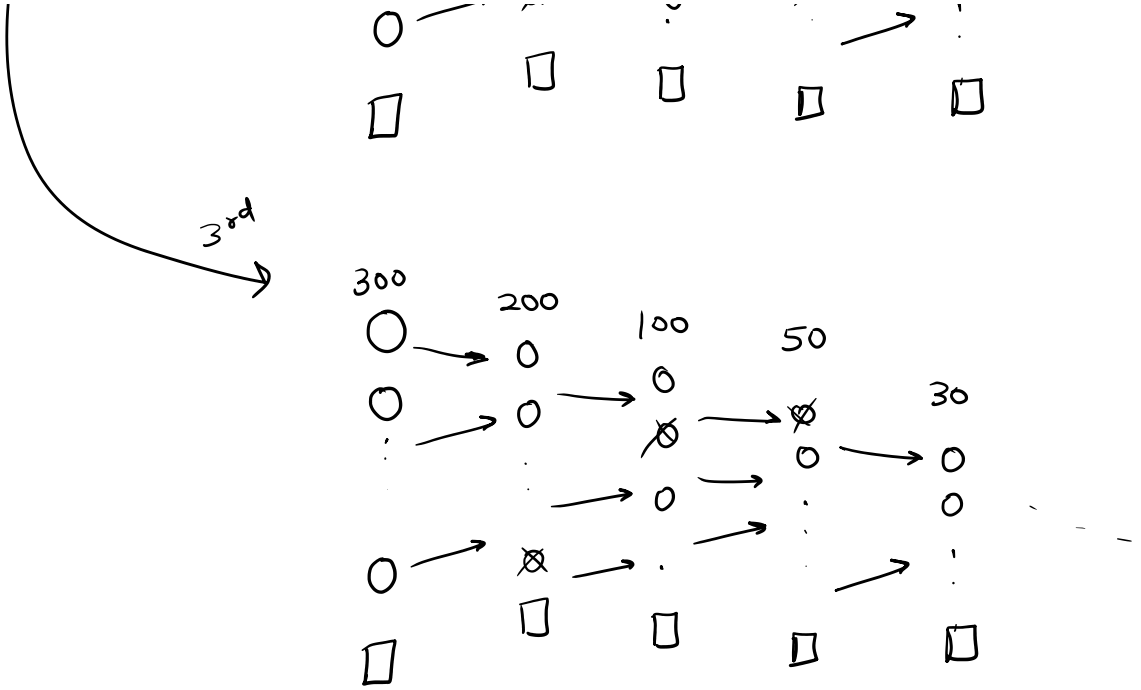
Ordinary Gradient Descent : $\underset{wt}{new} := \underset{wt}{old} - \eta \frac{\partial J}{\partial w}$

Momentum Optimizer :-

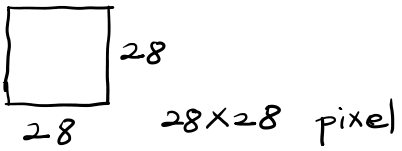
$$m := \beta m - \eta \frac{\partial}{\partial w} J(w)$$
$$w := w + m$$







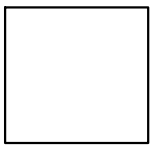
MNIST



0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Kernel / Filter

1	1	1
0	0	1
0	1	0

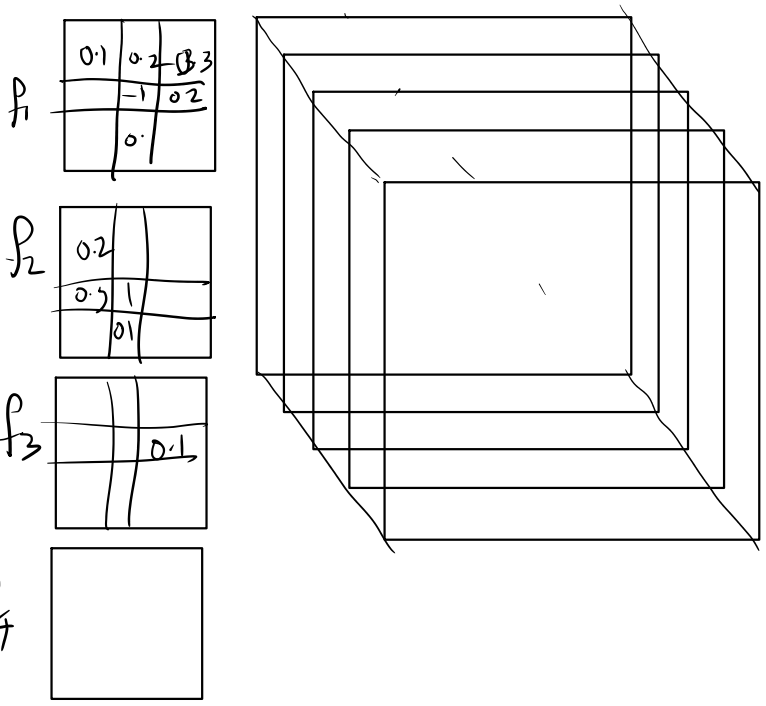


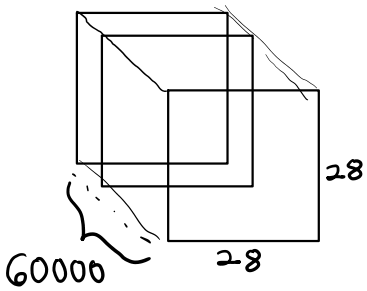
1	1	1	1	1	1	0
1	2	3	3	5	2	1
0	0	0	2	2	1	1
0	0	2	3	1	1	0
0	2	3	2	2	1	0
1	3	2	2	1	0	0
2	2	2	1	0	0	0

dot product

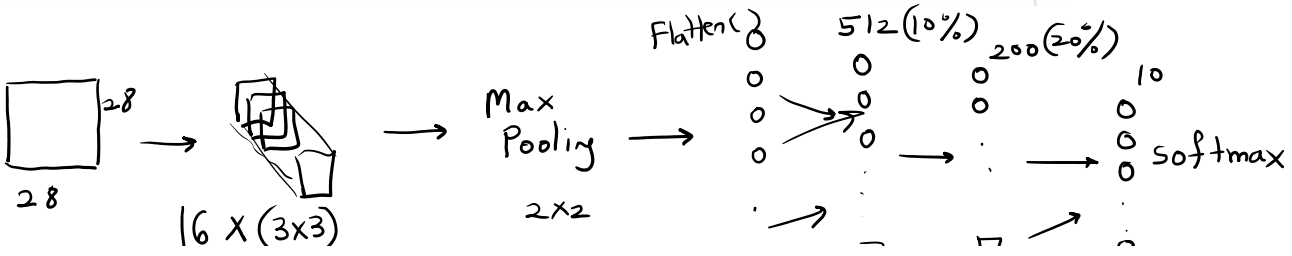
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

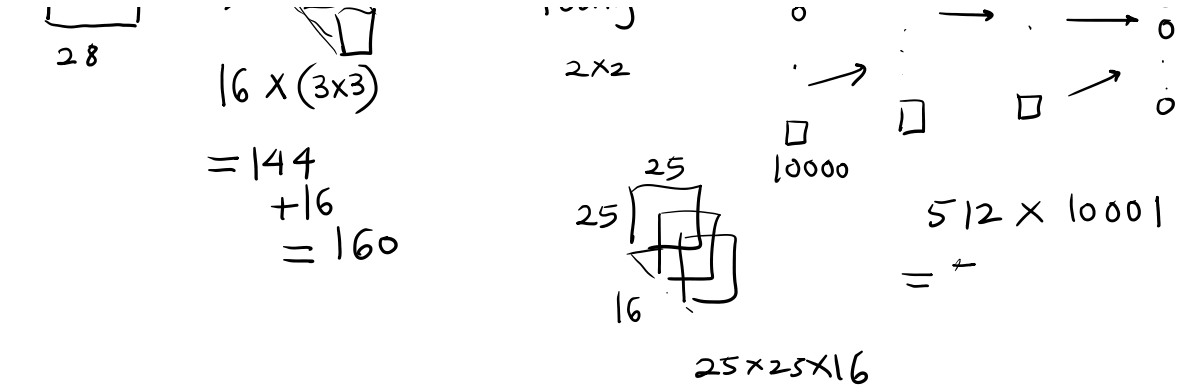
0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

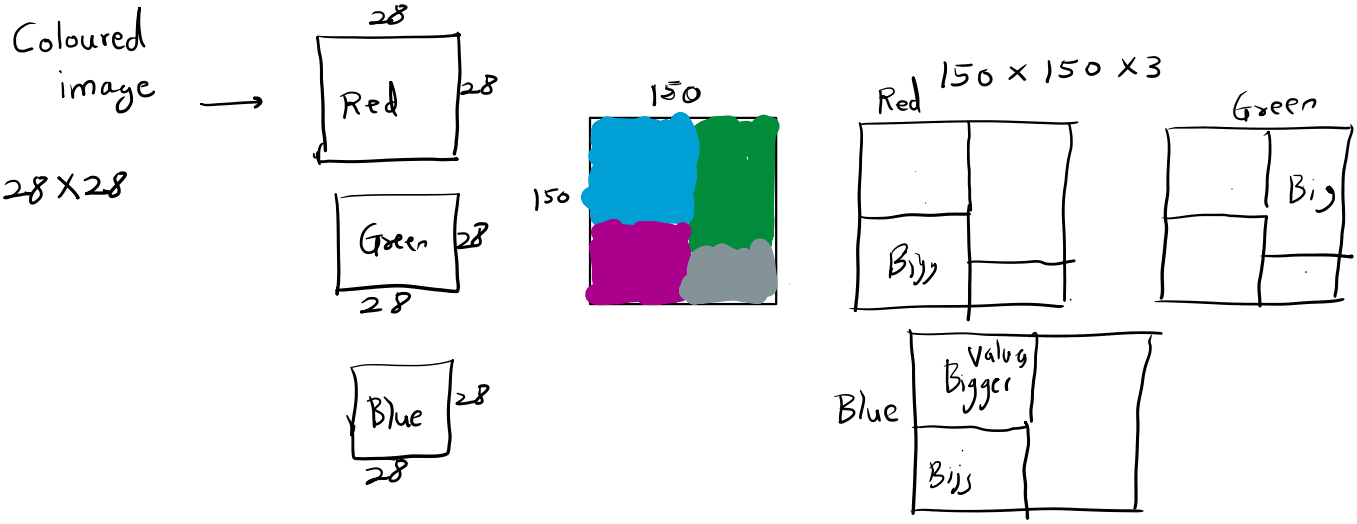


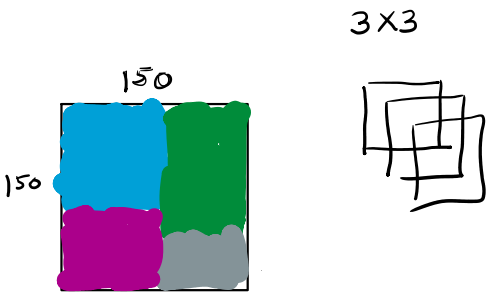


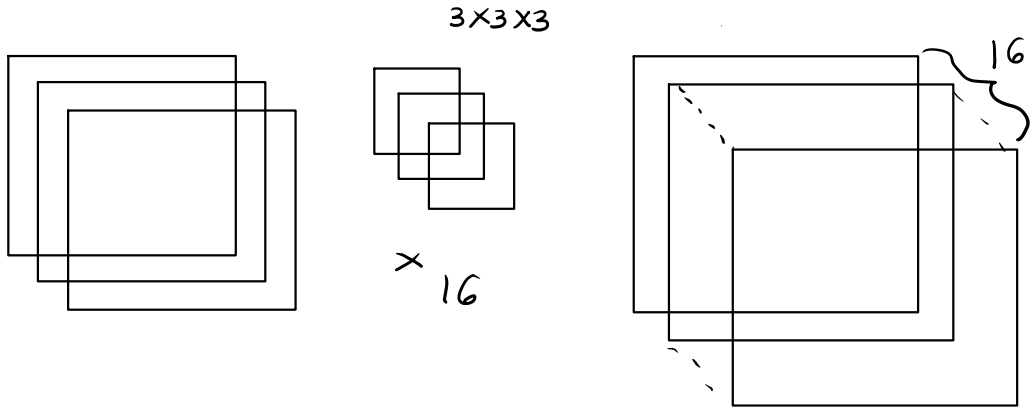

```
model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 28x28
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(28, 28, 1)),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2),strides=1),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(200, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
```











Data Augmentation: When the dataset images are less in number, we augment the

data by flipping, rotating or shifting

```
datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    shear_range=60,
    validation_split=0.2)
```