From DNN to CNN

Enhancing Computer Vision with Convolutions

Laurence Moroney, Google

```
mnist = tf.keras.datasets.fashion_mnist

(training_images, training_labels),
(val_images, val_labels) = mnist.load_data()

training_images=training_images / 255.0
val_images=val_images / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(20, activation=tf.nn.relu),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

```
mnist = tf.keras.datasets.fashion_mnist

(training_images, training_labels),
(val_images, val_labels) = mnist.load_data()

training_images=training_images / 255.0
val_images=val_images / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(20, activation=tf.nn.relu),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

```
mnist = tf.keras.datasets.fashion_mnist

(training_images, training_labels),
(val_images, val_labels) = mnist.load_data()

training_images=training_images / 255.0
val_images=val_images / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(20, activation=tf.nn.relu),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

```
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3053 - accuracy: 0.8885 - val_loss: 0.3849 - val_accuracy: 0.8645
Epoch 15/20
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3012 - accuracy: 0.8910 - val_loss: 0.3865 - val_accuracy: 0.8663
Epoch 16/20
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2979 - accuracy: 0.8923 - val_loss: 0.3766 - val_accuracy: 0.8690
Epoch 17/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2937 - accuracy: 0.8938 - val_loss: 0.3814 - val_accuracy: 0.8655
Epoch 18/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2917 - accuracy: 0.8936 - val_loss: 0.3897 - val_accuracy: 0.8647
Epoch 19/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2897 - accuracy: 0.8942 - val_loss: 0.3943 - val_accuracy: 0.8626
Epoch 20/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2865 - accuracy: 0.8953 - val_loss: 0.3846 - val_accuracy: 0.8677
<tensorflow.python.keras.callbacks.History at 0x7fa71c6eb5f8>
```

```
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3053 - accuracy: 0.8885 - val_loss: 0.3849 - val_accuracy: 0.8645
Epoch 15/20
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3012 - accuracy: 0.8910 - val_loss: 0.3865 - val_accuracy: 0.8663
Epoch 16/20
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2979 - accuracy: 0.8923 - val_loss: 0.3766 - val_accuracy: 0.8690
Epoch 17/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2937 - accuracy: 0.8938 - val_loss: 0.3814 - val_accuracy: 0.8655
Epoch 18/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2917 - accuracy: 0.8936 - val_loss: 0.3897 - val_accuracy: 0.8647
Epoch 19/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2897 - accuracy: 0.8942 - val_loss: 0.3943 - val_accuracy: 0.8626
Epoch 20/20
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2865 - accuracy: 0.8953 - val_loss: 0.3846 - val_accuracy: 0.8677
<tensorflow.python.keras.callbacks.History at 0x7fa71c6eb5f8>
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28,
1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

Your turn!