Curso: Visión por computadora – Sistemas computacionales – ITAM

Semestre enero-mayo 2019.

Paola Mejia (157093)

Práctica 6: Regresión con AutoEncoders Convolucionales

Denoising

7. Describe el mejor modelo que hayas encontrado: balance entre mejor desempeño y menos parámetros. Menciona el tiempo que toma el entrenamiento del modelo seleccionado.

```
_________________________________________________________________
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_17 (InputLayer)        (None, 64, 64, 3)         0         
_________________________________________________________________
batch_normalization_33 (Batc (None, 64, 64, 3)         12        
_________________________________________________________________
conv2d_62 (Conv2D)           (None, 64, 64, 8)         224       
_________________________________________________________________
max_pooling2d_21 (MaxPooling (None, 32, 32, 8)         0         
_________________________________________________________________
conv2d_63 (Conv2D)           (None, 32, 32, 16)        1168      
_________________________________________________________________
dropout_21 (Dropout)         (None, 32, 32, 16)        0         
_________________________________________________________________
up_sampling2d_21 (UpSampling (None, 64, 64, 16)        0         
_________________________________________________________________
conv2d_64 (Conv2D)           (None, 64, 64, 8)         1160      
_________________________________________________________________
batch_normalization_34 (Batc (None, 64, 64, 8)         32        
_________________________________________________________________
conv2d_65 (Conv2D)           (None, 64, 64, 3)         219       
=================================================================
Total params: 2,815
Trainable params: 2,793
Non-trainable params: 22
```

El tiempo promedio por época es 62s. El mejor desempeño se obtuvo después de 91 épocas.

El tiempo total de entrenamiento fue de 1 hora y media.

Es el mejor modelo porque la pérdida de validación es de 0.0031 y tiene 2,792 parametros.

El modelo tiene 4 capas convolucionales. La primera y la tercera tienen 8 filtros. La segunda incrementa el número de filtros a 16 y finalmente la última tiene 3 filtros.

Las entradas se normalizan utilizando Batch Normalization. A continuación, se disminuyen los datos con la operación max-pooling y después de una convolución con 16 filtros, la operación upsampling hace que vueltan a su tamaño original.

```
i = Input(x_train_noise.shape[1:])
h = BatchNormalization()(i)

h = Conv2D(8, (3, 3), padding='same', activation='elu')(h)
h = MaxPooling2D((2, 2))(h)

h = Conv2D(16, (3, 3), padding='same', activation='elu')(h)
h = Dropout(0.2)(h)

h = UpSampling2D((2, 2))(h)
h = Conv2D(8, (3, 3), padding='same', activation='elu')(h)

h = BatchNormalization()(h)
o = Conv2D(3, (3, 3), padding='same', activation='sigmoid')(h)

model = Model(inputs=i, outputs=o)
```

8. Reporta el proceso de minimización de la pérdida por época, del modelo seleccionado.

```
Train on 4000 samples, validate on 1000 samples
Epoch 1/200
4000/4000 [==============================] - 73s 18ms/step - loss: 0.0617 - val_loss: 0.0296
Epoch 2/200
4000/4000 [==============================] - 71s 18ms/step - loss: 0.0294 - val_loss: 0.0294
Epoch 3/200
4000/4000 [==============================] - 76s 19ms/step - loss: 0.0215 - val_loss: 0.0257
Epoch 4/200
4000/4000 [==============================] - 68s 17ms/step - loss: 0.0173 - val_loss: 0.0186
Epoch 5/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0144 - val_loss: 0.0142
```

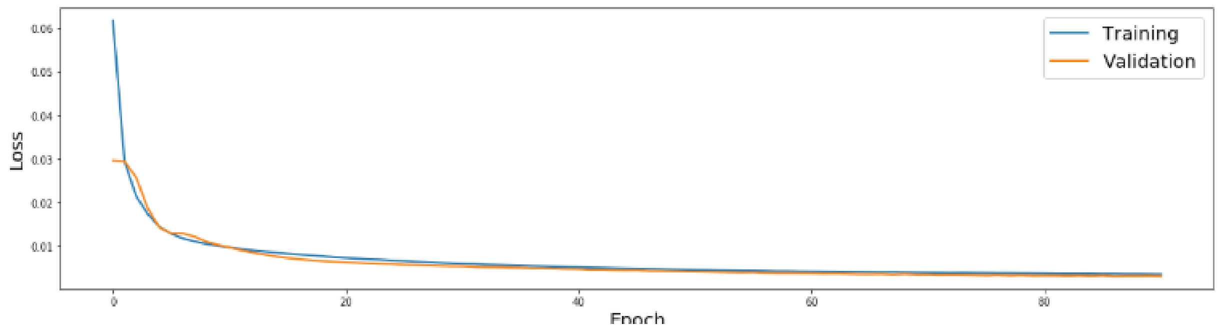
```
Epoch 6/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0128 - val_loss: 0.0129
Epoch 7/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0118 - val_loss: 0.0129
Epoch 8/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0111 - val_loss: 0.0122
Epoch 9/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0105 - val_loss: 0.0110
Epoch 10/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0101 - val_loss: 0.0103
Epoch 11/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0097 - val_loss: 0.0097
Epoch 12/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0094 - val_loss: 0.0090
Epoch 13/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0090 - val_loss: 0.0085
Epoch 14/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0088 - val_loss: 0.0081
Epoch 15/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0085 - val_loss: 0.0076
Epoch 16/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0083 - val_loss: 0.0073
Epoch 17/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0081 - val_loss: 0.0070
Epoch 18/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0079 - val_loss: 0.0068
Epoch 19/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0077 - val_loss: 0.0066
Epoch 20/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0075 - val_loss: 0.0064
Epoch 21/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0073 - val_loss: 0.0063
Epoch 22/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0071 - val_loss: 0.0062
Epoch 23/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0070 - val_loss: 0.0060
Epoch 24/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0069 - val_loss: 0.0059
Epoch 25/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0067 - val_loss: 0.0059
Epoch 26/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0066 - val_loss: 0.0057
Epoch 27/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0064 - val_loss: 0.0057
```

```
Epoch 28/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0063 - val_loss: 0.0056
Epoch 29/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0062 - val_loss: 0.0055
Epoch 30/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0061 - val_loss: 0.0054
Epoch 31/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0060 - val_loss: 0.0054
Epoch 32/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0059 - val_loss: 0.0052
Epoch 33/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0058 - val_loss: 0.0052
Epoch 34/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0057 - val_loss: 0.0051
Epoch 35/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0056 - val_loss: 0.0051
Epoch 36/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0055 - val_loss: 0.0050
Epoch 37/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0054 - val_loss: 0.0049
Epoch 38/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0054 - val_loss: 0.0049
Epoch 39/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0053 - val_loss: 0.0048
Epoch 40/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0052 - val_loss: 0.0047
Epoch 41/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0051 - val_loss: 0.0047
Epoch 42/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0051 - val_loss: 0.0046
Epoch 43/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0050 - val_loss: 0.0046
Epoch 44/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0049 - val_loss: 0.0045
Epoch 45/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0049 - val_loss: 0.0044
Epoch 46/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0048 - val_loss: 0.0044
Epoch 47/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0048 - val_loss: 0.0043
Epoch 48/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0047 - val_loss: 0.0042
Epoch 49/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0046 - val_loss: 0.0042
```

```
Epoch 50/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0046 - val_loss: 0.0042
Epoch 51/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0046 - val_loss: 0.0041
Epoch 52/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0045 - val_loss: 0.0041
Epoch 53/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0045 - val_loss: 0.0040
Epoch 54/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0044 - val_loss: 0.0040
Epoch 55/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0044 - val_loss: 0.0039
Epoch 56/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0043 - val_loss: 0.0040
Epoch 57/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0043 - val_loss: 0.0038
Epoch 58/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0042 - val_loss: 0.0038
Epoch 59/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0042 - val_loss: 0.0038
Epoch 60/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0042 - val_loss: 0.0037
Epoch 61/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0042 - val_loss: 0.0037
Epoch 62/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0041 - val_loss: 0.0037
Epoch 63/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0041 - val_loss: 0.0037
Epoch 64/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0041 - val_loss: 0.0036
Epoch 65/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0040 - val_loss: 0.0035
Epoch 66/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0040 - val_loss: 0.0035
Epoch 67/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0040 - val_loss: 0.0036
Epoch 68/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0040 - val_loss: 0.0034
Epoch 69/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0040 - val_loss: 0.0036
Epoch 70/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0039 - val_loss: 0.0034
Epoch 71/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0039 - val_loss: 0.0034
```

```
Epoch 72/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0039 - val_loss: 0.0034
Epoch 73/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0038 - val_loss: 0.0034
Epoch 74/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0039 - val_loss: 0.0034
Epoch 75/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0038 - val_loss: 0.0033
Epoch 76/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0038 - val_loss: 0.0033
Epoch 77/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0038 - val_loss: 0.0033
Epoch 78/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0038 - val_loss: 0.0032
Epoch 79/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0037 - val_loss: 0.0033
Epoch 80/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0037 - val_loss: 0.0032
Epoch 81/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0037 - val_loss: 0.0032
Epoch 82/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0037 - val_loss: 0.0032
Epoch 83/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0037 - val_loss: 0.0031
Epoch 84/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0036 - val_loss: 0.0032
Epoch 85/200
4000/4000 [==============================] - 63s 16ms/step - loss: 0.0036 - val_loss: 0.0031
Epoch 86/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0036 - val_loss: 0.0032
Epoch 87/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0036 - val_loss: 0.0031
Epoch 88/200
4000/4000 [==============================] - 62s 16ms/step - loss: 0.0036 - val_loss: 0.0031
Epoch 89/200
4000/4000 [==============================] - 61s 15ms/step - loss: 0.0036 - val_loss: 0.0031
Epoch 90/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0035 - val_loss: 0.0032
Epoch 91/200
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0035 - val_loss: 0.0031
Epoch 00091: early stopping
```

```
4000/4000 [==============================] - 62s 15ms/step - loss: 0.0035 - val_loss: 0.0031
Epoch 00091: early stopping
```

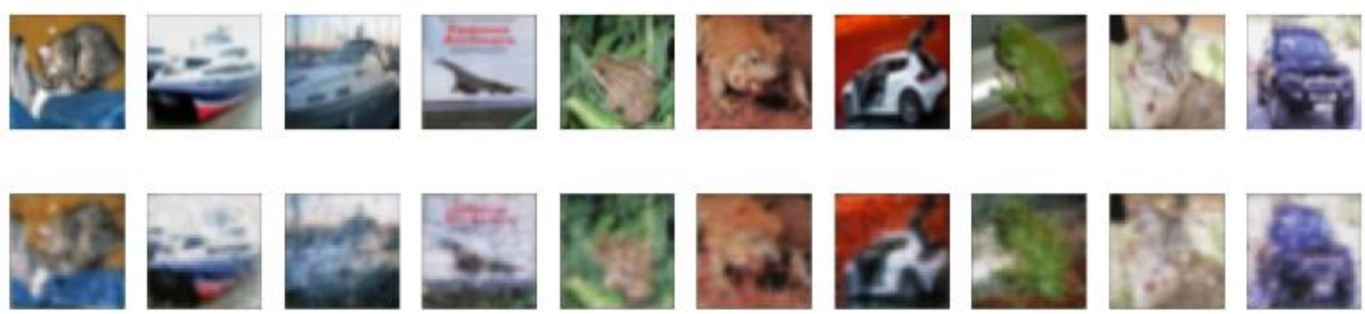


9. Presenta una tabla que compare el modelo seleccionado contra algunos de los otros modelos que hayas evaluado (los más prometedores). Dependiendo del problema que hayas decidido atacar, la comparación puede incluir tiempo de entrenamiento, pérdida final, error de clasificación.

	Pérdida de validación (Epoch 10)	**Diferencia entre pérdida de validación y de entrenamiento (sobre-ajuste)**	**Tiempo de entrenamiento (Epoch 5)**	**Número de parámetros entrenables**
Original (Filtros 8 y tamaño de batch 512)	0.0201	0.0070	76s	138,601
Aumentando número de filtros	0.0172	0.0070	716s	1,698,057
Aumentando tamaño de batch (batch size = 32)	0.0087	0.0013	555s	1,698,057
Aumentando tamaño de batch (batch size = 64)	0.0072	0.0006	546s	1,698,057
Aumentando tamaño de batch (batch size =128)	0.0066	0.0006	546s	1,698,057
Aumentando tamaño de batch (batch size =256)	0.0064	0.0006	554s	1,698,057
Quitando capa densa	0.0054	0.0008	1937s	426,447
Quitando capa densa y capas duplicadas de convoluciones (aún hay 6 capas convolucionales)	0.0048	0.0007	1170s	186,441
Simplificar modelo pasado. Borrar dos capas convolucionales. Total 4 capas convolucionales.	0.0067	0.0002	271s	38,793

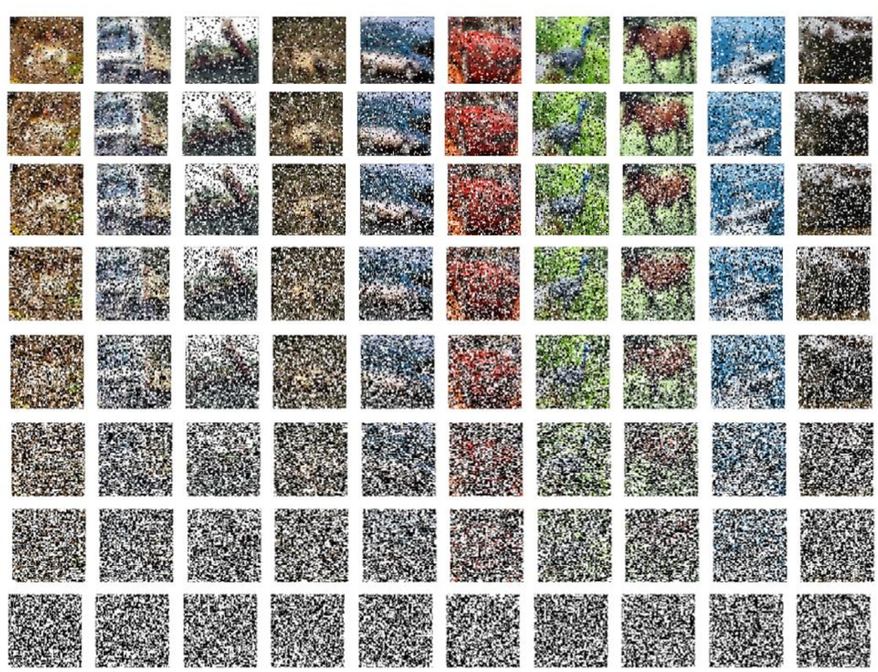
Simplificar modelo pasado reduciendo número de filtros (16,32,16)	0.0094	0.0013	397s	10,185
Simplificar modelo pasado reduciendo número de filtros aún más (8,16,8) [MODELO SELECCIONADO]	0.0121	0.0000	72s	2,793
Modelo pasado con 91 epocas	0.0031	0.0004	62s	2,793
Simplificar modelo pasado reduciendo número de capas convolucionales a 2.	0.0163	0.0087	69s	921
Simplificar modelo pasado reduciendo número de capas convolucionales a 1.	0.0593	0.0109	31s	96

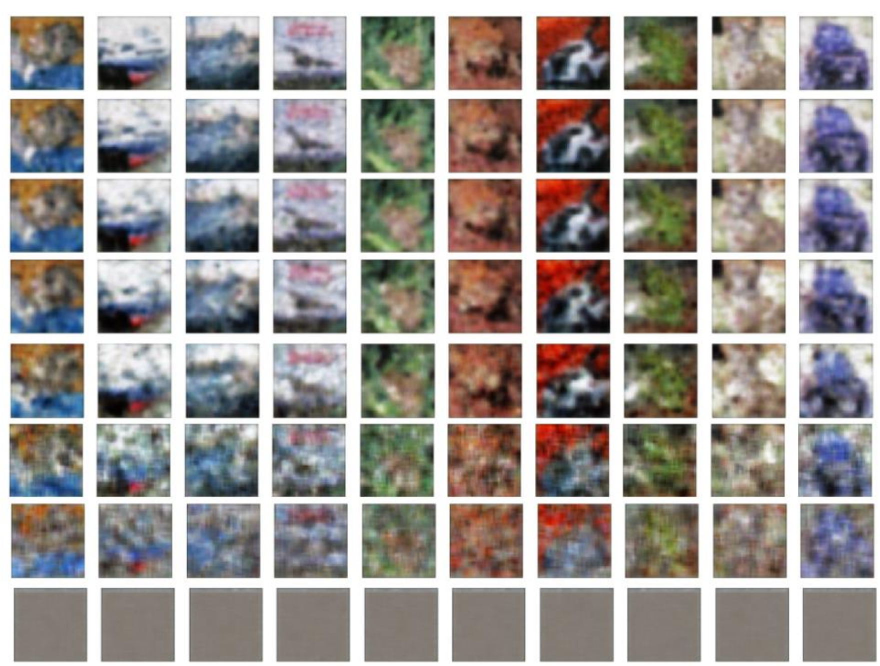
10. Presenta ejemplos de la reconstrucción o predicción de imágenes para los casos de denoising o segmentación, o la matriz de confusión para el caso de extracción de representaciones.



11. Variación de ruido.

El original usa el 20% de ruido. A continuación, se muestran las imágenes originales con mayor porcentaje de ruido y su respectiva reconstrucción.





Salt & Pepper	MSE
20%	0.0031
30%	0.0040
40%	0.0052
50%	0.0073
60%	0.0107
80%	0.0275
90%	0.0447
100%	0.0594

Tabla mostrando el error cuadrático medio con mayores porcentajes de ruido