

Proyecto #3: Deep Learning CNN
Maestría en ciencia de la computación
Prof. Cristian López Del Alamo

Para este proyecto se pide aplicar un modelo *Deep Learning* basado en CNN con el objetivo de clasificar imágenes de rayos X de pulmón. El *dataset*, tiene información de 4 grupos de imágenes. Grupo de imágenes de pacientes con COVID, pacientes con radiografías de pulmón normales, con opacidad de pulmón y finalmente con Neumonía Viral.

Su equipo deberá modificar la estructura propuesta en clase y encontrar los hiperparámetros del modelo. Finalmente, deberán realizar los experimentos adecuados a fin alcanzar un *accuracy* mayor a 85 %

Nota: El objetivo es modificar la estructura propuesta en clase, la cual se detalla a continuación.

```
1 num_classes = 10
2 learning_rate = 0.001
3 num_epochs = 20
4
5
6 class CNN(nn.Module):
7     def __init__(self, num_classes=10):
8         super(CNN, self).__init__()
9         self.layer1 = nn.Sequential(
10             nn.Conv2d(in_channels=1, out_channels=16, kernel_size=3, stride=1, padding=2),
11             nn.ReLU(),
12             nn.MaxPool2d(kernel_size=2, stride=2))
13         self.layer2 = nn.Sequential(
14             nn.Conv2d(16, 32, kernel_size=5, stride=1, padding=2),
15             nn.ReLU(),
16             nn.MaxPool2d(kernel_size=2, stride=2))
17         self.fc = nn.Linear(7*7*32, num_classes)
18
19     def forward(self, x):
20         out = self.layer1(x)
21         out = self.layer2(out)
22         out = out.reshape(out.size(0), -1)
23         out = self.fc(out)
24         return out
25
```

Actividades

1. Utilizar la **Base de Datos** y analizar su información.
2. Separar el *dataset* en 70 % para entrenamiento, 20 % para validación y 10 % para testing. Recuerde que los datos deben ser tomados de forma completamente *random*.
3. Modificar el modelo incrementando el número de capas. En el ejemplo anterior sólo hay dos capas.

4. Analizar lo que ocurre cuando se añade *batch normalization* a cada capa.
5. Analizar lo que ocurre cuando se añade *batch normalization* y *Dropout* a cada capa.
6. Analizar que ocurre si sólo se agrega *Dropout* y *batch normalization* a la primera capa.
7. Pruebe utilizando *EarlyStopping*, grafique la función de pérdida.
8. Utilice 70 % para entrenamiento y el resto para *testing*.
9. Su grupo deberá preparar una presentación de 15 minutos y exponer su solución en clase.

Fecha de entrega: Domingo 5 de diciembre hasta las 12 de la noche

Cada equipo deberá subir un sólo documento con la siguiente estructura:

1. Introducción.
2. Explicación: Explicar la arquitectura de o de los modelos seleccionados.
3. El documento deberá ser desarrollado estilo *paper* y teniendo en cuenta el siguiente formato *ieee*
4. Experimentos: Deberá explicar qué y cómo se hicieron los experimentos, además de colocar sus resultados utilizando tablas y gráficos que soporten sus experimentos.
5. Conclusiones: Deberá redactar las conclusiones de acuerdo a los resultados.

Incluir, en el documento, el *link* al *github* para verificar el código fuente.

Nota: Toda copia parcial o total en el documento o código fuente invalida en trabajo de todo el equipo.