

# Predicción de energía eólica con redes neuronales de convolución

Ignacio Fernández Sánchez-Pascuala

Javier Castellano Soria

*Universidad Complutense de Madrid | Universidad Politécnica de Madrid*

2 de marzo de 2024

## Resumen

La energía eólica es una fuente de energía renovable variable en la que el conocimiento a futuro de la potencia que aportará a la red eléctrica es crucial para su integración efectiva. El objetivo de este trabajo es mostrar la capacidad que poseen las redes neuronales de convolución para predecir la producción de energía eólica en una región. Esto se ejemplificará con el ajuste de una arquitectura de este tipo para el pronóstico de datos ofrecidos por Red Eléctrica sobre la producción horaria total de todos los parques de Mancha Oriental. Se detallarán las variables y recursos de los que se disponen así como las ventajas que ofrecen los distintos sesgos inductivos considerados en el procesamiento de datos y la construcción del modelo.

## 1. Introducción

Uno de los grandes problemas al que se enfrenta la energía eólica, es su incorporación de manera efectiva al suministro eléctrico debido a su variabilidad difícil de predecir. El mercado eléctrico debe mantener un equilibrio entre oferta y demanda para evitar posibles apagones o daños en equipos eléctricos. Esto en ocasiones se solventa teniendo centrales de energía no renovable preparadas para suplir cualquier cambio repentino en la producción de los aerogeneradores, lo cual supone una barrera para obtención de energía limpia y renovable. Además, las penalizaciones que sufren las empresas energéticas por la incorrecta programación de energía que pretenden vender provocan un encarecimiento de la misma entre otros factores.

Por este motivo, en este trabajo se trata de mostrar la eficiencia de las redes neuronales de convolución para la obtención de pronósticos a partir de datos meteorológicos. Primero se mostrará el problema específico que se plantea, luego se explicarán las variables empleadas y la arquitectura de red neuronal de convolución, y finalmente se mostrarán los resultados.

## 2. Enunciado del problema

El problema consiste en la predicción de la potencia media que va a ofrecer una unidad de programación (parque o conjunto de parques eólicos) en una determinada hora. En concreto, se dispondrá de un histórico desde 2018 hasta 2021 (ambos años incluidos) con dicha potencia por cada hora generada por todos los parques de Mancha Oriental. Es decir, la potencia asociada a la hora  $t$ , será la media a lo largo de la franja  $[t - 1, t]$  de la suma de las potencias dadas por todo el sistema de aerogeneradores. La finalidad es, dada una hora a futuro, predecir esa potencia media a partir de unas determinadas variables y un modelo adecuado que explicaremos a continuación.

## 3. Variables y recursos.

Las variables regresoras que se emplearán serán las componentes  $u$  y  $v$  de la velocidad del viento (en km/h) a 100 metros de la superficie, datos facilitados por la empresa RAVENWITS. La elección de esta variable meteorológica se debe a que los aerogeneradores poseen una altura parecida e incluso superior en los más actuales. En concreto, por cada hora  $t$  se tiene una matriz  $60 \times 60$ , con dimensiones latitud y longitud respectivamente, con dos canales. El elemento  $ij$  de la matriz sería el par de componentes de la velocidad del viento  $(u_{ij}^t, v_{ij}^t)$  en la ubicación con latitud  $i$ -ésima y longitud  $j$ -ésima. De esta forma, se tiene un mallado con una precisión de  $0'1^\circ$  de toda la región de Mancha Oriental con la información del viento en cada punto. En la Figura 1 se muestra de forma clara cómo son los datos.

LAT\LONG	-5'0°	-4'9°	-4'8°	-4'7°
49'3°	$(u_{11}, v_{11})$	$(u_{12}, v_{12})$	$(u_{13}, v_{13})$	...
49'2°	$(u_{21}, v_{21})$	$(u_{22}, v_{22})$	$(u_{23}, v_{23})$	...
49'1°	$(u_{31}, v_{31})$	$(u_{32}, v_{32})$	$(u_{33}, v_{33})$	...
49'0°	...	...	...	...

Figura 1: Matriz  $60 \times 60$  con dos canales que son las componentes  $u$  y  $v$  del viento en las coordenadas correspondientes en una determinada hora.

Estos mapas corresponden a las predicciones meteorológicas del viento más recientes del modelo HRES del *European Centre for Medium-Range Weather Forecasts* (ECMWF). El ECMWF ofrece una previsión del tiempo a medio plazo que se va actualizando a las 00:00 y a las 12:00 de cada día. Por ejemplo, el dato de los vientos a las 02:00 y el de los vientos a las 11:00 provienen de la predicción realizada a las 00:00 del mismo día. Cabe destacar que, una vez entrenado el modelo, si quisiéramos predecir la producción dentro de dos días, por ejemplo, no estaríamos considerando la predicción meteorológica más reciente que se podría obtener en ese mismo día y se cometería un mayor error al incorporar el meteorológico. Esto se podría disminuir tomando como datos extra los vientos predichos con mayor tiempo de antelación, pero aumentaría el coste computacional y no es el objetivo obtener un modelo tan refinado.

## 4. Procesado de datos

A partir de los datos explicados anteriormente, se realizaron una serie de transformaciones previas con la esperanza de que nuestro modelo interprete de manera más eficaz los datos.

La potencia máxima real en vatios en el eje del rotor de un aerogenerador es [1]:

$$P = cte \cdot r^2 \cdot \rho^3 \quad (1)$$

con  $cte$  una constante que depende de la densidad del aire y el tipo de rotor,  $r$  la longitud en metros del aspa y  $\rho$  el módulo de la velocidad del viento en km/h. Esta relación se puede comprobar en la Figura 2 donde se muestra un mapa de calor con la correlación entre  $\rho^3$  de cada punto del mallado con la producción. Además, las zonas con mayor correlación, corresponden con la presencia de mayor densidad de parques eólicos en el centro del mapa.

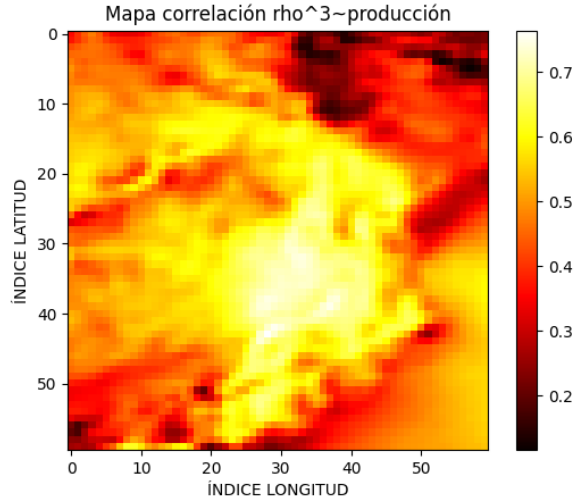


Figura 2: Mapa de calor  $60 \times 60$  donde el píxel  $ij$  tiene un color en función de la correlación del histórico del módulo de la velocidad del viento al cubo de coordenadas latitud  $i$ -ésima y longitud  $j$ -ésima, con la producción total.

Debido a esta relación directa del módulo con la potencia, se decidió transformar las componentes  $u$  y  $v$  en coordenadas polares  $\rho$  y  $\theta$ . Sin embargo, trabajar con el ángulo directamente supone una dificultad a nuestro modelo al no recoger la noción de que un ángulo cercano a  $0^\circ$  y otro a  $360^\circ$  son realmente datos similares. Por ello, se decidió codificar el ángulo considerando  $\sin(\theta)$  y  $\cos(\theta)$ . De esta forma se obtiene un mapa con tres canales  $\rho$ ,  $\sin(\theta)$ ,  $\cos(\theta)$  según:

$$\rho = \sqrt{u^2 + v^2} \quad | \quad \sin(\theta) = \frac{v}{\rho} \quad | \quad \cos(\theta) = \frac{u}{\rho}. \quad (2)$$

Por otro lado, la variable objetivo que refleja la potencia media ofrecida por todo el sistema durante la hora anterior a una dada, se transformó dividiendo por la potencia máxima instalada. Esta cantidad indica la máxima potencia que puede ofrecer toda la región. Así, la variable

objetivo, tras multiplicarla por 100, va a reflejar el rendimiento en porcentaje por cada hora. Esto se realiza para poder comparar medidas con otras unidades de programación que no posean la misma instalación.

Resumiendo, nuestro conjunto de datos va a consistir, para cada hora entre los años 2018 y 2021, en un par formado por un mapa de vientos (entrada) de dimensión  $60 \times 60 \times 3$  que corresponde a una matriz con los tres canales mencionados, y por el rendimiento en porcentaje (salida). Nótese que para predecir el rendimiento en una hora  $t$  solo se tiene en cuenta los vientos en esa hora  $t$  aun cuando la producción se refiere a una medida tomada a lo largo del intervalo  $[t - 1, t]$ . Podríamos haber considerado en vez del mapa en tiempo  $t$ , el de tiempo  $t - 1$ . Por este motivo, es más preciso alimentar la red con ambos mapas para que tenga información de la meteorología en  $t - 1$  y  $t$ , pero requeriría un mayor gasto de memoria y recursos computacionales.

## 5. Modelo de redes neuronales de convolución

Las redes convolucionales han sido usadas satisfactoriamente en diversos contextos como la clasificación de imágenes. Su arquitectura es capaz de extraer características de datos que presentan una topología de malla. Esta es la razón por la que usaremos este tipo de modelos de redes neuronales para capturar patrones en los mapas de viento que nos permitirá predecir la producción de energía. Esto se consigue gracias a la operación de convolución y la de reducción de dimensionalidad, *pooling*, que explicaremos a continuación.

### 5.1. Convolución

Para entender esta técnica vamos a mostrar el caso de dos dimensiones. Supongamos que tenemos como entrada una matriz  $M$  de dimensión  $n \times m$ , a la que vamos a aplicar una convolución, y otra  $K$  de dimensión  $r \times s$  llamada *kernel* con  $r < n$  y  $s < m$ . Sea  $N$  la convolución de  $M$  vía el kernel  $K$ , entonces:

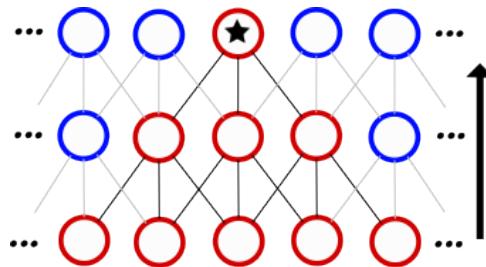
$$N(i, j) = \sum_{l=1}^r \sum_{k=1}^s K(l, k) \cdot M(i + l - 1, j + k - 1) \quad (3)$$

con  $i$  tal que  $i + r - 1 \leq n$  y  $j$  tal que  $j + s - 1 \leq m$ . Es decir,  $i = 1, \dots, n - r + 1$  y  $j = 1, \dots, m - s + 1$ . La nueva matriz obtenida  $N$  se conoce como *mapa de características* a la que se le suele aplicar una función de activación. Los parámetros de nuestra red neuronal serían los términos del kernel.

En la práctica se suele añadir un término de sesgo como parámetro que se suma a todos los elementos de  $N$ . Es decir, se usaría un solo parámetro de sesgo para cada mapa de características.

$$N(i, j) = \sum_{l=1}^r \sum_{k=1}^s \sum_{p=1}^c K(l, k, p) \cdot M(i + l - 1, j + k - 1, p) \quad (4)$$

Una de las motivaciones para usar esta operación de convolución como procesamiento de la entrada frente a la estructura de PCM es las escasas interacciones. En una capa de una PCM se realiza un producto por una matriz de parámetros donde cada uno describe la interacción entre una unidad de entrada y otra de salida. Sin embargo, el número de parámetros e interacciones es menor en una convolución. Esto implica computacionalmente una mejora en memoria y número de operaciones. Además, aunque haya menos interacciones, las unidades de capas más profundas pueden interactuar de forma indirecta con gran parte de la entrada como se muestra en la figura 3.



## 5.2. Pooling

5

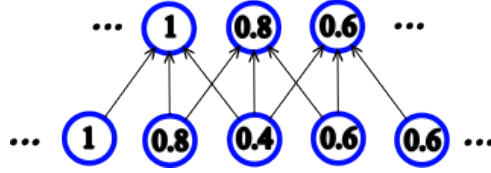


Figura 4: Ejemplo de *pooling* tomando el máximo de tamaño 3 en una dimensión.

Esta técnica se aplica a la matriz en cuestión con la finalidad de reducir la dimensión de la entrada resumiendo la información de varias unidades vecinas en una sola. Para esto, se aplica el *pooling* de tal forma que cada unidad solo participe en una operación. Por ejemplo, si tenemos un vector de dimensión  $3n$  y aplicamos un *pooling* como el de la Figura 4, pero esta vez cada tres unidades, se obtendrá un vector que resumirá la información del primero con dimensión  $n$ . Esto se realiza análogamente en el caso caso bidimensional considerando como unidades cercanas parches  $r \times s$  con saltos de  $r$  y  $s$  en las filas y las columnas respectivamente.

De esta forma, primero la capa de convolución con la compartición de parámetros, permite la detección de un patrón en cualquier parte del mallado ya que el kernel se aplica a cada zona de la matriz con los mismos parámetros. Después, al aplicar un *pooling* se consigue resumir la información en un espacio de dimensión menor que se interpretará como las características de la entrada. Generalmente se emplean varias capas de convolución y *pooling* para la obtención de dicho espacio, después se anida al modelo una PCM para predecir o clasificar a partir del espacio de menor dimensión.

Cabe destacar que este tipo de modelos ofrece cierta invarianza local debido a la compartición de parámetros y el uso de técnicas de *pooling* (véase Figura 4 si el vector de debajo se desplaza una posición). Esto para la detección y clasificación de imágenes resulta una ventaja ya que aporta invarianza bajo traslación. Sin embargo, esto no supone una mejora en el problema a tratar pues no se produce la misma energía si unos determinados vientos se encuentran en una zona u otra. Esto hace reflexionar acerca de posibles mejoras que podría aportar variaciones como las *Visual Transformer* (ViT) [2], o las *Graph Neural Network* (GNN) [3] que han resultado ser exitosas en la predicción meteorológica (*GraphCast* [4]).

### 5.3. Modelo considerado

El modelo que se ha construido sigue un esquema similar al presentado en numerosas arquitecturas para la clasificación de imágenes como Le-Net [5] (véase Figura 5), una de las primeras redes de convolución; o Alex-Net [6], una red más compleja que supuso una gran mejora en la clasificación de imágenes del conjunto *ImageNet*. Primero se aplican capas que consisten en una convolución aumentando la cantidad de mapas de características seguida de un *pooling* para reducir la dimensión extrayendo la información de nuestro mapa de vientos. Después se aplica una PCM cuya salida unidimensional será la predicción de la producción.

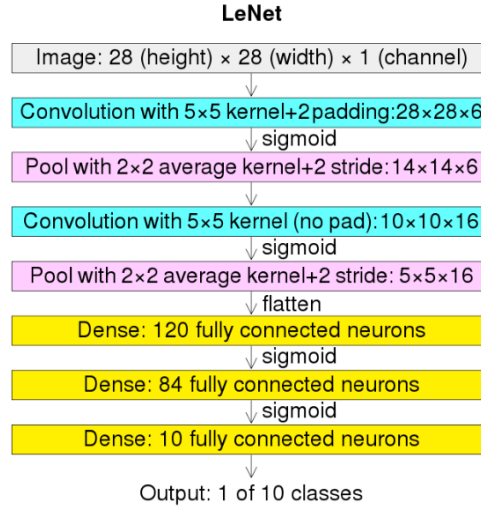


Figura 5: Diagrama de la red neuronal de convolución LeNet. Figura extraída de [7].

Se ha empleado la función de activación ReLU con ecuación (5) puesto que en la actualidad es una de las funciones más empleadas. Esto se debe a que suele presentar un aprendizaje más rápido porque su derivada, con ecuación (6), no se anula para valores altos a diferencia de otras funciones sigmoides que tienden a ser cero provocando que el gradiente se anule.

$$\text{ReLU}(x) = \begin{cases} 0, & \text{si } x < 0 \\ x, & \text{si } x \geq 0 \end{cases} \quad (5)$$

$$\text{ReLU}'(x) = \begin{cases} 0, & \text{si } x < 0 \\ 1, & \text{si } x > 0 \end{cases} \quad (6)$$

En la Figura 6 se muestra la arquitectura considerada con las dimensiones de salida de cada capa y los parámetros empleados. Primero se realiza la extracción de características alternando tres capas de convolución de tamaño  $5 \times 5$  con 6, 16 y 32 kernels respectivamente con tres capas de *pooling* de tamaño  $2 \times 2$ . Después, el nuevo espacio de características  $4 \times 4 \times 32$  se aplanan para tener un vector de dimensión 512 que será la entrada de una PCM de una capa oculta de 100 neuronas. La salida del modelo también tiene activación ReLU ya que nuestros datos son mayores que 0. Como las predicciones deben estar entre 0 y 100 al tratarse de porcentajes, una predicción mayor que 100 debe truncarse a 100 y tenerse en cuenta ante una posible sobre carga.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 56, 56, 6)	456
max_pooling2d (MaxPooling2D)	(None, 28, 28, 6)	0
conv2d_1 (Conv2D)	(None, 24, 24, 16)	2416
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 16)	0
conv2d_2 (Conv2D)	(None, 8, 8, 32)	12832
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 32)	0
flatten (Flatten)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 100)	51300
dropout_1 (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 1)	101
=====		
Total params: 67105 (262.13 KB)		
Trainable params: 67105 (262.13 KB)		

Figura 6: Red neuronal de convolución propuesta.

La cantidad de parámetros empleada es 67105, muy reducida a si se hubiese empleado una PCM con una capa oculta. En este último caso, si dicho espacio latente tuviese 512 neuronas como en el modelo propuesto, se tendría un total de  $60 \cdot 60 \cdot 3 \cdot 512 + 512 + 512 \cdot 1 + 1 = 5.530.625$  parámetros. Esto es un punto a favor a considerar un modelo de convolución con una complejidad menor y que es eficiente extrayendo características para la posterior predicción.

## 6. Entrenamiento

Para el entrenamiento adecuado del modelo se llevó a cabo una gestión inteligente de los datos para controlar el sobreajuste. El error a minimizar se consideró la media del valor absoluto del error debido a su fácil interpretación para posibles consideraciones que pudiera tomar una compañía eléctrica tras los resultados obtenidos. Para el proceso de optimización se utilizó el algoritmo adam.

### 6.1. Gestión inteligente de los datos

Los datos que engloban los años 2018 hasta 2021 ambos incluidos, se dividieron en entrenamiento, años 2018 y 2019; validación, año 2020; y test, año 2021. Conviene señalar que los datos al tratarse de una serie temporal no son independientes. Datos de horas cercanas tendrán valores similares, por tanto, si hubiésemos hecho una partición aleatoria estaríamos cometiendo un error en la estimación del error. Esto se debe a que un sobreajuste en entrenamiento podría derivar a un error muy bajo en test ya que es probable que un dato del test esté cercano en tiempo



a otro del train para el que se sobreajustó el modelo obteniendo un error menor al real. Por este motivo, la partición se realizó de tal forma que los datos estén lo suficientemente separados.

El conjunto de entrenamiento será utilizado para minimizar el error mientras que el de validación se empleará para determinar en qué momento parar el proceso de optimización, esto se conoce como *Early Stopping*. Si el algoritmo de optimización minimiza correctamente, a medida que aumenta el número de épocas se espera que la función de coste en entrenamiento se reduzca continuamente, sin embargo, esto no implica que el modelo mejore su capacidad de generalización. Por este motivo, en cada época se evalúa la pérdida en validación, si el error en este no ha mejorado en una cantidad de épocas (paciencia) el proceso de entrenamiento termina al considerar que no se puede mejorar la generalización si se continúa optimizando.

## 6.2. Optimización

Para el ajuste de los parámetros se tomó como función de error la media de los errores absolutos como se comentó anteriormente. Como algoritmo de optimización se eligió el algoritmo adam (*adaptive momentum*) tomando *mini-batches* de 64 datos. Es uno de los algoritmos de optimización más empleados en el contexto del aprendizaje automático para la búsqueda de un mínimo óptimo. Se basa en el descenso del gradiente, pero aporta una serie de modificaciones en la actualización de los pesos que permite, en un marco general, una mejor convergencia. Por un lado, incorpora un momento de orden 1 que permite simular la inercia de un objeto al ir por una pendiente acelerando su convergencia y escapando de mínimos locales. Por otro, incorpora un momento de orden 2 con el que regula la tasa de aprendizaje para cada parámetro con la finalidad de tomar una dirección de aprendizaje más directa al mínimo.

El cálculo del gradiente se consigue con retropropagación del error al igual que en el caso de una PCM usando la regla de la cadena.

También se consideró regularización *dropout* de 0,2 en la capa de entrada y en la oculta de la PCM final. Esta técnica consiste en apagar, fijar a cero, de forma aleatoria el 20 % de las neuronas en cada iteración del entrenamiento forzando al modelo a realizar la predicción usando menos parámetros. Una vez entrenado, las predicciones se realizan con todas las neuronas.

## 7. Resultados

En la Figura 7 se muestra la curva de aprendizaje. Se observa cómo el error en entrenamiento va reduciéndose a medida que se aumentan las épocas. Por otro lado, en validación sigue un trazado más inestable, en la época número 20 se consigue el mejor resultado y después de 10 épocas no mejora parando el entrenamiento y quedándose el modelo con los pesos de la época número 20.

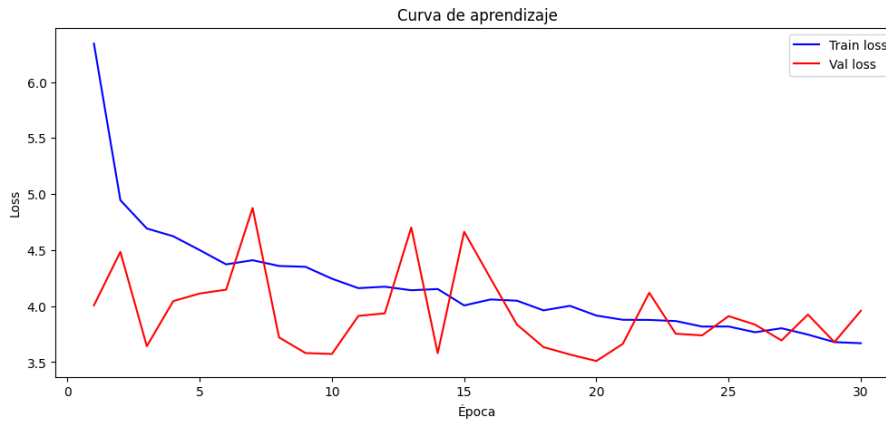


Figura 7: Curva de aprendizaje.

En la evaluación del error en el conjunto test, 2021, se obtuvo un valor de 3,62 mucho menor a 16,10 error asociado a considerar como predicción la media de todas las producciones.

## 8. Conclusión y reflexión sobre los aspectos éticos y contribución hacia los ODS

Como se ha podido comprobar con el modelo que se ha considerado, se consiguen resultados prometedores para la predicción de energía eólica. Un control sobre la producción a futuro, permite a las energéticas una mejor gestión de sus recursos planificando adecuadamente el suministro energético. Esto supone un beneficio inmenso para las mismas a la vez que debería traducirse en un abaratamiento del precio de la energía.

De esta forma, el objetivo final de una predicción precisa debe estar encaminado a la reducción de la pobreza y de la desigualdad garantizando un suministro energético a la mayor parte de la población. A su vez, facilita la inclusión de energías renovables como la eólica al mercado, apostando por una energía no contaminante.

## Agradecimientos

Se agradece a Red Eléctrica por los permisos concedidos para el uso de su histórico de producciones en Mancha Oriental. También se agradece a la empresa RAVENWITS S.L. por la cesión de las variables meteorológicas usadas del viento a 100 metros.

## Referencias

- [1] [https://www.mapa.gob.es/ministerio/pags/Biblioteca/Revistas/pdf\\_REA %20FREA\\_1980\\_04\\_97\\_100.pdf](https://www.mapa.gob.es/ministerio/pags/Biblioteca/Revistas/pdf_REA %20FREA_1980_04_97_100.pdf)
- [2] Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. *Advances in Neural Information Processing Systems*, 34, 15908-15919.
- [3] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- [4] Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., ... & Battaglia, P. (2022). GraphCast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*.
- [5] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324. <https://doi.org/10.1109/5.726791>
- [6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of The ACM*, 60(6), 84-90. <https://doi.org/10.1145/3065386>
- [7] Wikipedia contributors. (2023). LeNet. *Wikipedia*. <https://en.wikipedia.org/wiki/LeNet>