
Redes neuronales para la medición de entrelazamiento cuántico con datos incompletos

Javier A. Murillo David Vanegas Edgar E. Navas Joar E. Buitrago

Departamento de Física

Universidad Nacional de Colombia,

Carrera 45 No. 26-85, Edificio Uriel Gutiérrez, Bogotá D.C., Colombia.

Resumen

Este proyecto fue la reproducción del trabajo de Koutny et al. [14], el cuál explora la aplicación de redes neuronales artificiales en la predicción de la concurrencia de estados cuánticos entrelazados a partir de datos incompletos. Se desarrollaron dos modelos de redes neuronales: uno entrenado a partir de una cantidad específica de proyectores y otro entrenado a partir de datos completos pero que es capaz de recibir valores incompletos como entrada y aún así realizar predicciones. Estos modelos se crearon con la librería keras de python y se entrenaron para predecir la concurrencia de estados cuánticos utilizando matrices de densidad generadas a partir de matrices aleatorias. El entrenamiento se llevó a cabo principalmente aprovechando la potencia computacional de Kaggle. Sin embargo, debido a la falta de la potencia computacional requerida para la tarea se decidió limitar el estudio al caso de estados entrelazados de 2 qubits y únicamente a la predicción de la concurrencia. Se encontraron similitudes importantes en nuestros resultados en comparación con los resultados de Koutny et al. En el comportamiento de el error medio absoluto en función del número de proyectores comprobando que para números bajos de proyectores el error crece, mientras que para valores altos el error permanece muy bajo. Se llegó a que la red con mayor precisión en el conjunto completo de pruebas que se realizó fue la llamada measurement specific, mientras tanto la red llamada measurement independent presenta errores del mismo orden de la measurement specific para algunas pruebas mientras que para otras produce errores del mismo orden que los producidos por el algoritmo MaxLik.

Palabras Clave Inteligencia Artificial, Entrelazamiento Cuántico, Concurrencia, Redes Neuronales.

1. Introducción

La presente investigación se lleva a cabo debido al crecimiento en el desarrollo de técnicas de inteligencia artificial en los últimos años, específicamente su aplicación en la física [12, 19, 2]. Las redes neuronales artificiales son un grupo de modelos dentro del área de investigación de la inteligencia artificial que se basan en crear conexiones entre objetos llamados perceptrones; a estos objetos también se les llama neuronas debido al símil que se hace usualmente entre la red neuronal biológica de los seres vivos y las redes neuronales artificiales. El conjunto de varias capas de neuronas conectadas entre ellas que reciben un input y devuelven un valor o un conjunto de valores siguiendo determinadas funciones se denomina red neuronal [9].

Se le ha encontrado uso a las redes neuronales en la resolución de problemas muy complejos o que requieren un poder computacional muy grande para llegar a una solución exacta. Las redes neuronales sacrifican la exactitud de la solución a cambio de una velocidad de predicción órdenes de magnitud mayor [22]. Las redes neuronales son capaces de hacer predicciones con respecto a problemas que ni siquiera tienen solución exacta. El proceso de entrenamiento de una red neuronal puede llevar desde minutos hasta meses, pero una vez entrenada es un modelo capaz de realizar predicciones respecto al tipo de problemas en los cuales se la entrenó en cuestión de segundos o minutos. Debido a esto, se ha comenzado a

utilizarlas en la ciencia para realizar predicciones en distintas áreas, como la astrofísica [23], los sistemas complejos [28], la física de partículas [17] y la física cuántica [24].

La aplicación de redes neuronales en el campo de la mecánica cuántica lleva algunas décadas en desarrollo y se está volviendo cada vez más popular debido a la facilidad con la que las redes neuronales son capaces de predecir características que usualmente requieren de cientos de mediciones experimentales, debido a la naturaleza probabilística de la mecánica cuántica [20], tal es el caso de la tomografía cuántica que pretende reconstruir un estado cuántico, que en su propia naturaleza es probabilístico, a partir de medir repetidas veces las distribuciones de probabilidad estados idénticos [11].

Es importante aclarar que este proyecto hace uso de redes neuronales clásicas para predecir características de estados cuánticos; por otro lado, el campo de las redes neuronales cuánticas es un campo aparte y requiere del uso de computadores cuánticos, lo cual se encuentra fuera de nuestro alcance.

2. Objetivo

Como se mencionó anteriormente, las redes neuronales son una herramienta capaz de hacer predicciones a partir de lo que está a «aprendido» durante su entrenamiento, entonces resulta coherente preguntarse si sería posible entrenar una

red neuronal a predecir la concurrencia de un estado cuántico entrelazado, ya que para la medición experimental de esta cantidad se requiere, en principio, de una tomografía completa del estado lo cual es considerablemente más costoso que entrenar una red neuronal; si fuese posible entrenar una red neuronal para hacer esto a partir de mediciones incompletas y sus resultados fuesen lo suficientemente confiables, el investigador de esta área del conocimiento tendría una herramienta muy poderosa a su lado, ya que le permitiría obtener una predicción confiable a partir de pocas mediciones. El propósito de este proyecto fue entrenar un conjunto de redes neuronales a predecir la concurrencia de estados cuánticos a partir de medidas incompletas.

La investigación realizada por Koutny et al. Tenía como objetivo principal ser lo más abarcadora posible. Esto significa entrenar redes neuronales para prever la concurrencia de cualquier estado cuántico entrelazado. Siguiendo esta misma línea, hemos decidido adoptar el mismo enfoque en nuestro proyecto; por lo tanto, el estado físico en el cual se enfoca esta investigación es el de cualquier sistema cuántico entrelazado de 2 qubits de información, esto incluye, por ejemplo, el caso de dos fotones con estados polarizados entrelazados [6] pero no se queda ahí, sino que se intenta incluir todo tipo de estados que cumplan estas condiciones. Para ello se presta especial atención en el apartado de la generación de datos 4.1 y en la prueba final utilizando estados de Werner 5.3.

3. Conceptos previos

Para el desarrollo de este proyecto es necesario el conocimiento previo de algunos conceptos, que permitirán contextualizar la investigación. En este apartado se examinarán algunos de ellos:

3.1. Estados cuánticos

Un estado cuántico codifica la información de un sistema físico, y cada sistema está asociado a un espacio de Hilbert, que es un espacio vectorial complejo con producto escalar e interno [25]. Hay dos tipos de estados de interés en la mecánica cuántica:

Los estados cuánticos puros son aquellos en los que un sistema se encuentra en un estado específico y bien definido. Estos estados están representados por vectores de estado únicos en el espacio de Hilbert y se pueden describir completamente con una función de onda o un ket, como $|\psi\rangle$. En un estado cuántico puro, se conoce con certeza la propiedad que se está midiendo, como la posición o el momento de una partícula [20].

Por otro lado, los estados cuánticos mixtos son una combinación probabilística de varios estados cuánticos puros representados por medio de una matriz de densidad ρ , que se define como:

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|,$$

Donde p_i son las probabilidades de encontrar el sistema en los estados cuánticos puros $|\psi_i\rangle$. Los estados mixtos representan situaciones en las que no podemos conocer con certeza

el estado exacto del sistema, lo que resulta en una descripción estadística [25]

3.2. Tomografía cuántica

La tomografía clásica es la obtención de una reconstrucción espacial de un objeto a partir de imágenes de sus selecciones [18]. La tomografía cuántica pretende utilizar este mismo concepto para encontrar la forma de un estado cuántico antes de las mediciones. Este proceso se debe a que, una vez se mide un estado cuántico, este colapsa en uno de los posibles resultados y queda definido; esto no es igual para el estado antes de las mediciones, el cual se encuentra en una superposición cuántica [26].

En general, la tomografía es un proceso muy demorado por la gran cantidad de mediciones. En el contexto cuántico se plantea reconstruir la función de onda de un sistema entrelazado, en el cual por cada partícula aumenta la cantidad de mediciones de forma exponencial, lo cual lo hace un proceso muy costoso [16].

3.3. Concurrencia

La concurrencia es una forma de medir el entrelazamiento cuántico en un sistema de 2 qubits. Varía de 0 a 1 donde un valor de concurrencia 0 representa un estado completamente separable, es decir, no entrelazado, mientras que un valor de 1 representa un estado completamente entrelazado en el cual ninguna característica del sistema puede describirse independientemente de ambos qubits [25]. Se utiliza la entropía de Shannon $S(\rho)$ con autovalores $\lambda_i(\rho)$ para crear esta medida. La entropía de Von Neumann también se emplea en la determinación del entrelazamiento de formación $E_F(|\psi\rangle)$ a través de las reducciones del estado puro $|\psi\rangle$ [25]. La concurrencia $C(\rho)$ en un sistema mixto de dos qubits (2×2) se define como $C(\rho) = \max(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4)$, donde λ_i son los autovalores de la matriz $\sqrt{\sqrt{\rho}\tilde{\rho}\sqrt{\rho}}$ en orden descendiente [10].

3.4. Los estados de Werner

En mecánica cuántica los estados de Werner son una clase de estado cuántico mixto que se puede representar como una matriz de dos estados superpuestos, estos estados fueron introducidos por Werner Heisenberg en 1927, se usan para sistemas cuánticos entrelazados [6]. Para dos qubits el estado de Werner se escribe de la forma:

$$\rho W(p) = p\rho\psi^- + \frac{1-p}{4}I. \quad (1)$$

Los estados de Werner son una buena prueba para nuestras redes debido a que la forma de la concurrencia de un estado de Werner se conoce teóricamente (2) y se puede usar para comprobar la predicción de nuestras redes [14],

$$C_W = \max[0, (3p - 1)/2]. \quad (2)$$

3.5. Redes neuronales

Las redes neuronales son un subconjunto de *machine learning* cuya estructura está inspirada en la forma en que las neuronas biológicas del cerebro humano comparten información.

Como su nombre lo indica, están compuestas por unidades de procesamiento llamadas neuronas, interconectadas en una estructura de capas que, al igual que las neuronas en el cerebro, reciben entradas, realizan un cálculo y producen una salida. Cada entrada tiene un peso que controla su contribución al cálculo de la neurona [27].

El proceso de aprendizaje de una red neuronal consiste en procesar los datos de entrada, hacer predicciones y ajustar los pesos cuando las predicciones son incorrectas, este proceso se repite muchas veces durante el entrenamiento hasta que la red mejora sus predicciones. Inicialmente, la ponderación de los pesos es aleatoria y las respuestas de la red pueden ser imprecisas, sin embargo, a medida que avanza el entrenamiento, la red ajusta gradualmente sus ponderaciones para replicar con mayor precisión los resultados.

3.5.1. ¿Cómo funciona una red neuronal?

En el corazón de una red neuronal se encuentra el perceptrón, la unidad básica de procesamiento. El perceptrón toma las entradas, las pondera utilizando pesos y sesgos asignados, y las suma. Posteriormente, aplica una función de activación a la suma ponderada para generar una salida. Esta función de activación introduce no linealidad en la red, lo que significa que la relación entre las entradas y las salidas no es simplemente una línea recta. Esto permite que la red aprenda patrones más complejos y representaciones jerárquicas en los datos. Finalmente, la salida del perceptrón se utiliza como entrada para otras neuronas en la red o para realizar predicciones si es la última capa.

El aprendizaje en las redes neuronales involucra una función de pérdida, que compara las predicciones del modelo con las respuestas correctas en los datos de entrenamiento. El objetivo es reducir esta discrepancia, mejorando así la precisión de las predicciones de la red. Para lograr esto, se ajustan los pesos y sesgos de la red para minimizar la pérdida. Este proceso se conoce como «backpropagation», realizado por los optimizadores de la red. Estos calculan el gradiente de la función de pérdida para cada peso y luego ajustan los pesos en la dirección que disminuye la pérdida[15].

La estructura fundamental de una red neuronal se organiza en capas, a continuación se presentan los 3 tipos básicos de capas que se usaron en el desarrollo de este proyecto, seguidos de 2 capas específicas para una red convolucional:

- **Capa de Entrada (Input Layer):** Se encarga de recibir los datos de entrada. Cada neurona en esta capa representa una característica de la cual la red deberá aprender y transmite esta información a las capas subsiguientes.
- **Capa Oculta (Hidden Layer):** Esta se encuentra entre la capa de entrada y la capa de salida y es responsable de realizar transformaciones y cálculos intermedios en los datos. Puede constar de una o varias capas, y la cantidad de neuronas en cada capa oculta puede variar según la complejidad del problema.
- **Capa de Salida (Output Layer):** La capa de salida produce los resultados finales de la red neuronal, que pueden ser valores continuos en problemas de regresión o

probabilidades en problemas de clasificación. La estructura y función de esta capa dependen de la naturaleza del problema que la red esté resolviendo.

- **Capa de Convolución (Convolutional Layer):** Este tipo de capa se utiliza comúnmente para procesar datos en formato de imagen. Aplica filtros de convolución para extraer características locales de la imagen como bordes o texturas. También se puede utilizar para dar un grado más de abstracción a la red, no solo para análisis de imagen.
- **Capa de Pooling (Pooling Layer):** Las capas de pooling se utilizan para reducir el tamaño espacial de las características extraídas por las capas de convolución. Esto ayuda a reducir la cantidad de parámetros en la red y a mejorar la invariancia a pequeñas variaciones en la posición de las características.

3.5.2. Las redes neuronales convolucionales

Las CNN, por sus siglas en inglés para Convolutional Neural Networks, son un tipo de red neuronal particularmente adecuada para tareas de procesamiento de imágenes, estas funcionan aplicando filtros a las imágenes para extraer características locales, como bordes y formas, lo que les permite reconocer objetos en imágenes, incluso si están rotadas, escaladas o traducidas [4].

El proceso de aprendizaje es iterativo y requiere múltiples ciclos (épocas) de entrenamiento. Con cada ciclo, la red ajusta sus pesos gradualmente, mejorando sus predicciones. A medida que continúa este proceso, la red neuronal aprende a reconocer patrones y, una vez entrenada, es capaz de realizar predicciones acertadas para conjuntos de datos nuevos.

4. Procedimiento

4.1. Generación de datos

Con el fin de representar los estados cuánticos con los que se va a entrenar la red se utiliza una matriz de densidad, la cual se crea a partir de dos matrices aleatorias U y G , también llamadas: la matriz del ensamble circular unitario (U) y la matriz de Ginibre (G) [14],

$$\rho = \frac{(1 + U^\dagger)GG^\dagger(1 + U)}{\text{Tr}[(1 + U^\dagger)GG^\dagger(1 + U)]}. \quad (3)$$

Esta matriz de densidad permite describir con generalidad estados cuánticos mixtos con un nivel de entrelazamiento aleatorio y debido a esto nos da una forma de crear estados cuánticos aleatorios en la cantidad necesaria para entrenar la red. Esta matriz de densidad es generada a partir de la medida de Haar [5], el cual es un concepto dentro de la teoría de grupos, la topología y la teoría de la medida que nos permite crear estados cuánticos aleatorios espaciados uniformemente entre ellos en un espacio de probabilidad, de esta manera nos aseguramos de que al entrenar la red con estos datos se la está entrenando de la forma más general posible [1] y va a ser capaz de hacer predicciones de datos reales arbitrarios. El producto GG^\dagger da como resultado una matriz semi-definida

positiva, mientras que el producto izquierdo y derecho, con los operadores $(1 + U^\dagger)$ y $(1 + U)$, respectivamente, representa una transformación unitaria, la cual distribuye los estados uniformemente siguiendo la medida de Haar, la división por la traza $\text{Tr}[(1 + U^\dagger)GG^\dagger(1 + U)]$ es el factor de normalización que asegura que la traza de la matriz de densidad sea 1.

Adicionalmente, el rango de la matriz G determina el número de estados puros de los que está compuesta, la superposición que es el estado mixto ρ (3), esto significa que cuando el rango de la matriz es 1 se tiene un estado puro, mientras que si el rango es distinto de uno se tiene un estado mixto de un número diferente de componentes.

Por otro lado, los observables en mecánica cuántica no son los estados en sí como matrices de densidad, sino las distribuciones de probabilidad del sistema proyectado, es por esto que se necesita hacer uso de los operadores de proyección para obtener las distribuciones de probabilidad para cada posible medición a partir de la matriz de densidad que representa el estado [20]. La forma de calcular el valor esperado de la matriz de densidad para un operador de proyección viene dada por la ecuación (4) [7].

$$p_m = \text{Tr}(\rho M_m), \quad (4)$$

Donde p_m es el valor esperado y M_m es el operador de proyección sobre el estado m .

4.2. Redes neuronales

Se desarrollaron dos modelos de redes neuronales usando Keras en Python. El primero se denomina «measurement-specific» y el segundo «measurement-independent». Ambos modelos tienen una estructura similar que incluye una capa de entrada, seis capas ocultas (densas) y una capa de salida. La única diferencia entre ellos es que el segundo modelo incorpora una capa convolucional y una de pooling adicionales a sus seis capas ocultas. La estructura interna de estas redes se ilustra en la figura 1.

El primer modelo de red neuronal, como su nombre indica, es específico para un número determinado de proyectores usados en el entrenamiento. Este modelo se entrena individualmente para cada número de proyectores, comenzando desde 36 y disminuyendo en pasos de 2 hasta llegar a 2 proyectores. En contraste, el segundo modelo se entrena solo una vez, utilizando información de los 36 proyectores. Si falta información de algún proyector, la red recibe ceros en su lugar.

En ambas redes, todas las capas utilizan la función de activación «relu» [3]. Esta elección es particularmente útil para la red «measurement-independent», ya que la función de activación «relu» ignora efectivamente los ceros recibidos, evitando que afecten las predicciones de la red.

Para medir el rendimiento de ambos modelos, se emplea el «Mean Squared Error» como función de pérdida. Esta es la función que la red busca minimizar, y se realiza a través del optimizador Nadam (Nesterov-accelerated Adaptive Moment Estimation) para ambos modelos [8]. Nadam es conocido por su capacidad para acelerar la convergencia y proporcionar un entrenamiento suave, lo cual nos será de gran utilidad debido a que manejaremos conjuntos de datos muy grandes para los entrenamientos, además de ser fuertemente valorado en redes neuronales profunda [13].

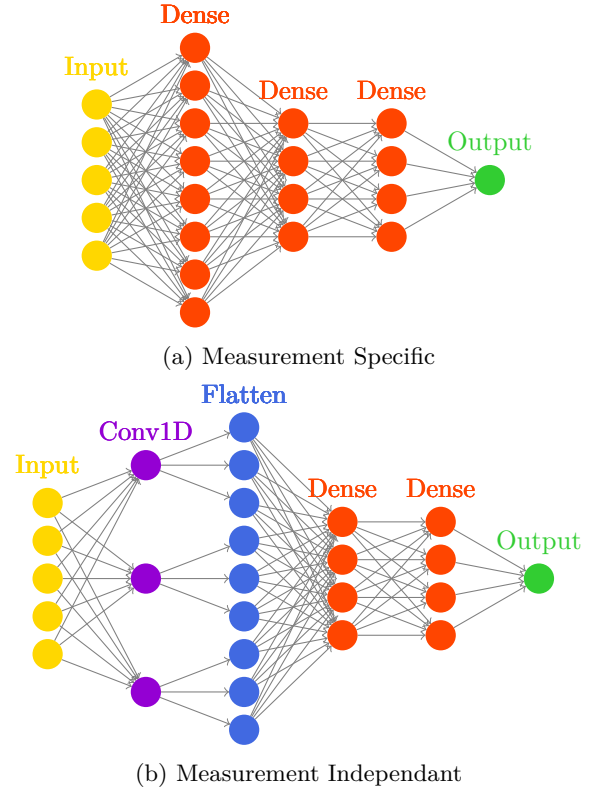


Figura 1: Estructura interna de las redes neuronales utilizadas. Las 6 capas densas de ambas redes han sido representadas como 2 con el propósito de simplificar la imagen.

4.3. Hardware usado

Utilizamos la potencia computacional de Kaggle, una plataforma web altamente recomendada por un compañero de clase y también por nosotros. Kaggle es conocida por ser la comunidad más grande de ciencia de datos, como menciona en su sitio web. Ofrece 30 horas semanales de acceso a tarjetas gráficas NVIDIA T4 Tensor Core y 16 GB de RAM, con un límite de uso continuo máximo de 12 horas. Para optimizar el proceso, implementamos cuatro cuentas en Kaggle. Cada red de «Measurement-Specific» requeriría aproximadamente de siete a ocho horas de cálculo ininterrumpido. En total, entrenamos 16 de estas redes, sumando alrededor de 120 horas de entrenamiento.

Por otro lado, la red «Measurement Independent» necesitaba unas 30 horas continuas de procesamiento. Esto nos llevó a recurrir a herramientas locales. Utilizamos un computador con un procesador intel I7 12950hx y una tarjeta gráfica GTX 4080 con 16 GB de RAM, que completó el entrenamiento en aproximadamente 35 horas a lo largo de 4 o 5 días.

5. Resultados

Las redes measurement specific fueron entrenadas con 900 mil datos, mientras que la red measurement independent fue entrenada con 7 millones. Una vez las redes han sido entrenadas, el siguiente paso es comparar los dos modelos de redes neuronales diseñados para este proyecto y el algoritmo Max-Lik. Utilizando un conjunto de datos de prueba, se elabora

un gráfico que muestra el Error Absoluto Medio (MAE) en relación con la cantidad de proyectores, donde las áreas sombreadas representan el espacio cubierto entre una desviación estándar y el valor medio del MAE. Este análisis se encuentra en la figura 2.

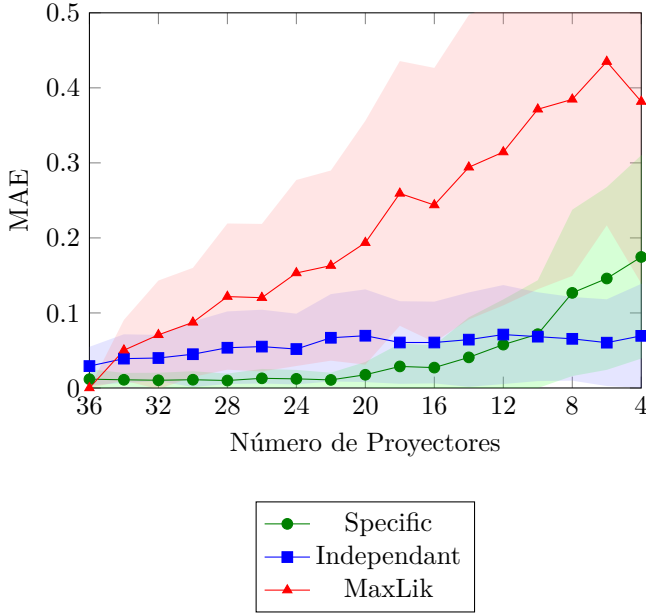


Figura 2: Gráfica de Error Absoluto Promedio (MAE) en función del número de proyectores para la red measurement independent, la measurement specific y el algoritmo MaxLik.

5.1. Measurement Specific

De color verde se observa en la figura 2 los datos obtenidos para el modelo Measurement Specific que se observa siempre por debajo del modelo de MaxLik y muy similar al modelo Measurement Independent. Sin embargo, para números de proyectores grandes fue el modelo con los errores más bajos entre todos los comparados. Por otro lado, para números de proyectores pequeños, por debajo de los 10 proyectores, su error empieza a crecer haciéndose más grande que el de segundo modelo de red entrenado en esta investigación, de esto se puede interpretar que la red measurement specific necesita entrenamiento adicional a la hora de trabajar con números de proyectores menores a 10.

5.2. Measurement Independent

En la figura 2 se distingue a los datos de la red Measurement Independent como los puntos, la línea y el área de color azul. En esta prueba, la red Measurement Independent muestra un comportamiento similar al de las redes Measurement Specific con un error siempre por debajo del error del algoritmo MaxLik, el cual es un estándar en la medición de propiedades como la concurrencia de estados cuánticos [21].

5.3. Estados de Werner

Finalmente, y como prueba final para los métodos de predicción, se les pidió un conjunto de predicciones de la concurrencia para los estados de Werner; lo interesante de esta

prueba es que la concurrencia de los estados de Werner se conoce teóricamente (sección 3.4) y representa una línea recta de pendiente $3/2$ a partir del valor $1/3$ (2). Esta prueba permite comprobar visualmente qué tan alejada se encuentra la predicción de los diferentes métodos de la curva teórica. El resultado se muestra en la figura 3.

La figura 3 muestra un excelente rendimiento de la red Measurement Specific, identificada en color verde. Sobresale consistentemente por su cercanía a la predicción teórica, superando a los otros dos métodos evaluados, incluso sin haber sido entrenada con estados de Werner. Este éxito sugiere que el método utilizado para separar estados a través de la medida de Haar es efectivo, proporcionando un entrenamiento uniforme y amplio que beneficia a la red, permitiéndole un alto rendimiento incluso en situaciones específicas como los estados de Werner. Además, este resultado resalta la precisión de este modelo por encima de los otros dos métodos utilizados.

La figura 3 presenta los resultados de la red Measurement Independent, en color azul. Debido a la limitación en la potencia computacional disponible para nuestro equipo, solo fue posible entrenar esta red con 7 millones de datos. A pesar de que en esta prueba la red Measurement Independent demostró ser la menos efectiva de las dos redes que entrenamos para este proyecto, su desempeño fue similar al del algoritmo MaxLik. Se cree que si se pudiese entrenar la red Measurement Independent con una mayor cantidad de datos, es probable que se logren errores más cercanos a los obtenidos por la red Measurement Specific, o al menos significativamente menores que los del algoritmo MaxLik.

6. Conclusiones

Este estudio se centró en la creación de dos tipos de redes neuronales para la predicción de la concurrencia en sistemas entrelazados de dos qubits. Para ello, se estudió y recopiló la teoría general sobre las redes neuronales, y se hizo un resumen de la teoría pertinente sobre la generación de los datos como matrices de densidad a partir de la teoría de matrices aleatorias. Para la parte principal de este proyecto, se crearon dos modelos de redes neuronales basadas enteramente en las redes creadas en la investigación de Koutny et al. [14]: las redes «measurement specific», entrenadas a partir de un número específico de proyectores, y la red «measurement independent», entrenada a partir de datos completos pero capaz de hacer predicciones con datos incompletos. Ambos modelos se entrenaron para recibir valores de la distribución de probabilidad del sistema proyectado y, con estos valores, predecir la concurrencia del sistema. Se encontró que las redes realizan constantemente predicciones más cercanas a la teórica en comparación con otros métodos usuales para calcular la concurrencia, como el algoritmo MaxLik. Finalmente, se probaron las redes con estados cuánticos de Werner como prueba visual de las predicciones de los tres métodos al variar el número de proyectores. En este caso, la red «measurement specific» se mantiene, para cada valor de proyectores, más cerca del valor de concurrencia teórica que los otros dos modelos. Por otro lado, la «measurement independent» realiza predicciones comparables a las del algoritmo MaxLik. Sin em-

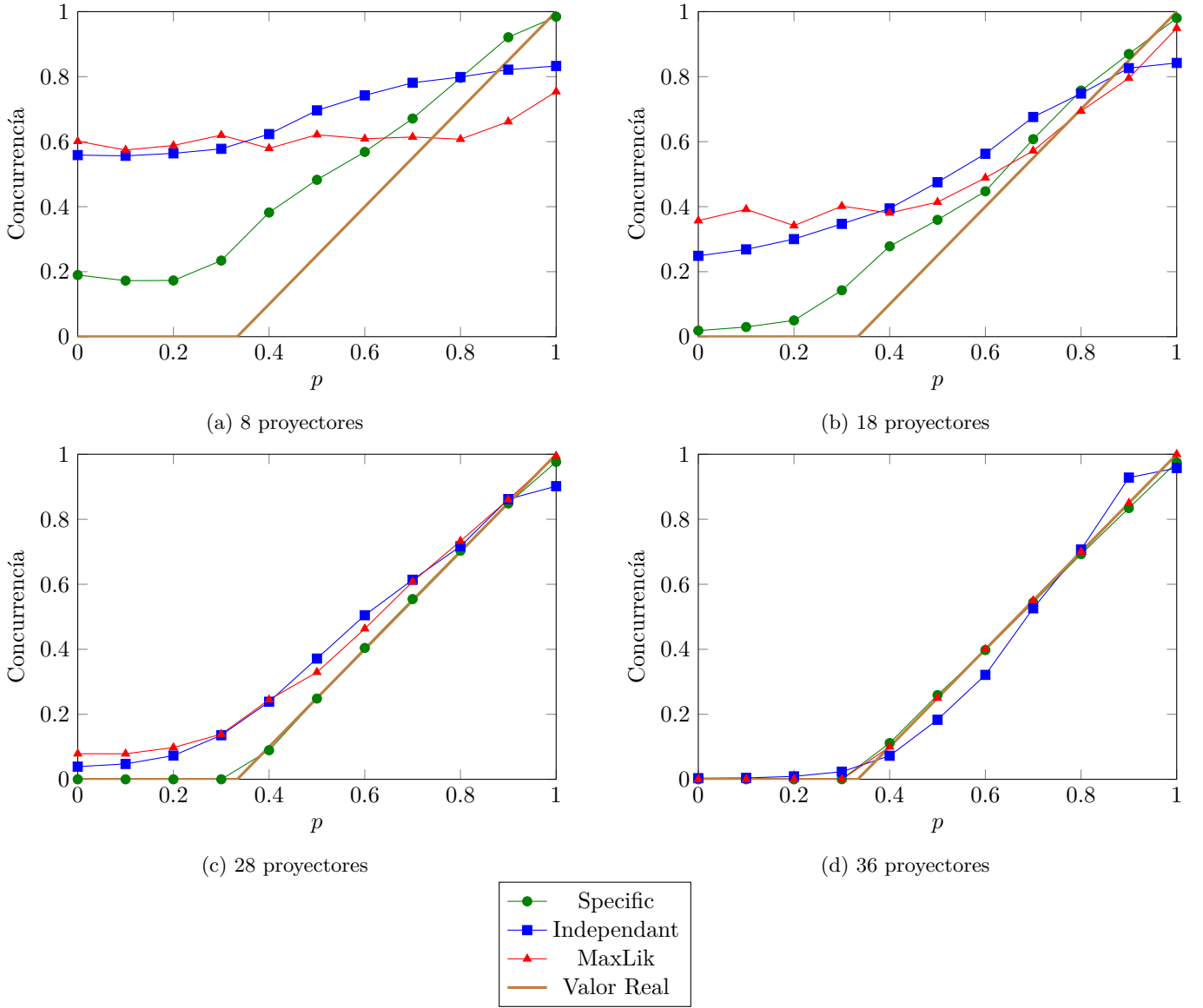


Figura 3: Gráfica de comparación entre los modelos measurement independent, measurement specific y MaxLik con el valor teórico de la concurrencia en función de parámetro p en un estado de Werner.

bargo, debido a la limitada capacidad computacional con la que contaba nuestro equipo, es de esperar que con un mayor entrenamiento y tamaño de redes se logren resultados mejores debido a que nuestras redes fueron una simplificación de las creadas por Koutny et al.

7. Reflexión final

7.1. Javier Alejandro Murillo

Me ha parecido uno de los ejercicios más provechosos que he hecho a lo largo de la carrera, ya que nos da una verdadera muestra de cómo es el trabajo dentro de la física; considero que a lo largo de la carrera se aprenden muchas cosas, pero no se aprende a ser físicos, el desarrollo de este proyecto y la retroalimentación del profesor me pareció fundamental para mi formación. Por otro lado, el proyecto me ha parecido de una relevancia enorme y agradezco mucho haber podido desarrollar este proyecto debido a que he aprendido de temas

con un enorme impacto en la investigación actual como lo son la inteligencia artificial que tiene un impacto enorme en todo el mundo y la medición de estados cuánticos que tiene una relevancia gigante dentro de la computación cuántica, ambos temas de último momento en la ciencia y tecnología.

7.2. David Vanegas Sanchez

En mi opinión, fue un curso bastante enriquecedor porque me enseñó cómo es el desarrollo de un proyecto de investigación de vanguardia de primera mano, así como la experiencia de realizar dicho proyecto con un grupo de trabajo muy diverso, donde cada integrante es versado en un campo diferente. Las exposiciones que realizamos a lo largo del curso son las que considero que tuvieron mayor impacto en mi experiencia del curso, ya que fueron muy diferentes a las exposiciones a las que se está acostumbrando durante la carrera y me llevaron a descubrir mis faltas en ese tipo de habilidades, vacíos conceptuales, y me impulsó a descubrir formas para abordar

estas falencias.

7.3. Edgar Eduardo Navas

En cuento a mi experiencia en la materia, la verdad ha sido enriquecedora, tanto por la adquisición de conocimiento cómo por el reconocer los vacíos que debo abordar para consolidar mi comprensión, identificando mis falencias como investigador y tratando de trabajar en ellas, transformándolas en oportunidades de conocimiento. Esta investigación me enseña que no solo se trata de adquirir conocimientos teóricos, sino también esta toma de conciencia me impulsa a abordar esos desafíos y convertirlos en oportunidades que contribuyen a mi crecimiento personal y profesional.

7.4. Joar Esteban Buitrago

Esta investigación ha revelado los vacíos teóricos presentes en mi comprensión de las ramas de la física cuántica. Agradezco a mis colegas por su colaboración, lo que me permitió enfrentar este desafío y obtener una comprensión más profunda de estos temas. Además, he tenido la oportunidad de compartir mis conocimientos en sistemas y programación, contribuyendo al desarrollo y comprensión de las redes por parte de mis compañeros. Este intercambio de conocimientos resalta la interdisciplinariedad inherente a todas las ramas del saber y subraya las aplicaciones prácticas que estas disciplinas tienen para abordar diversas problemáticas que surgen durante la investigación científica, logrando obtener resultados como los mostrados en el presente documento.

Referencias

- [1] George Biswas, Anindya Biswas y Ujjwal Sen. «Inhibition of spread of typical bipartite and genuine multi-party entanglement in response to disorder». En: *New Journal of Physics* 23.11 (nov. de 2021), pág. 113042. DOI: 10.1088/1367-2630/ac37c8.
- [2] Giuseppe Carleo et al. «Machine learning and the physical sciences». En: *Rev. Mod. Phys.* 91 (4 dic. de 2019), pág. 045002. DOI: 10.1103/RevModPhys.91.045002.
- [3] François Chollet et al. *Keras Activation Function*. 2015. URL: https://keras.io/api/layers/activation%5C_layers/relu/.
- [4] François Chollet et al. *Keras Convolutional Layers*. 2015. URL: https://keras.io/api/layers/convolution%5C_layers/.
- [5] D.L. Cohn. *Measure Theory*. Second. Birkhäuser Advanced Texts Basler Lehrbücher. Springer New York, 2013. ISBN: 9781461469568.
- [6] Artur Czerwinski. «Quantifying entanglement of two-qubit Werner states». En: *Communications in Theoretical Physics* 73.8 (jun. de 2021), pág. 085101. DOI: 10.1088/1572-9494/ac01e1.
- [7] Ivan B. Djordjevic. «Chapter 2 - Quantum Mechanics Fundamentals». En: *Quantum Information Processing, Quantum Computing, and Quantum Error Correction (Second Edition)*. Ed. por Ivan B. Djordjevic. Second Edition. Academic Press, 2021, págs. 31-95. ISBN: 978-0-12-821982-9. DOI: <https://doi.org/10.1016/B978-0-12-821982-9.00014-9>.
- [8] Timothy Dozat. «Incorporating Nesterov Momentum into Adam». En: *Stanford University* (2015).
- [9] Stephen Grossberg. «Nonlinear neural networks: Principles, mechanisms, and architectures». En: *Neural Networks* 1.1 (1988), págs. 17-61. DOI: 10.1016/0893-6080(88)90021-4.
- [10] Scott Hill y William K. Wootters. «Entanglement of a Pair of Quantum Bits». En: *Physical Review Letters* 78.26 (jun. de 1997), págs. 5022-5025. DOI: 10.1103/physrevlett.78.5022.
- [11] R. Horodecki et al. «Quantum entanglement». En: *Reviews of Modern Physics* 81.2 (2009), págs. 865-942. DOI: 10.1103/RevModPhys.81.865.
- [12] George Em Karniadakis et al. «Physics-informed machine learning». En: *Nature Reviews Physics* 3.6 (2021), págs. 422-440. DOI: 10.1038/s42254-021-00314-5.
- [13] Diederik P. Kingma y Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [14] Dominik Koutný et al. «Deep learning of quantum entanglement from incomplete measurements». En: *Science Advances* 9.29 (2023). DOI: 10.1126/SCIADV.ADD7131.
- [15] David Kriesel. *A Brief Introduction to Neural Networks*. 2007. URL: <http://www.dkriesel.com>.
- [16] Yi Li et al. «Two-Measurement Tomography of High-Dimensional Orbital Angular Momentum Entanglement». En: *Phys. Rev. Lett.* 130 (5 feb. de 2023), pág. 050805. DOI: 10.1103/PhysRevLett.130.050805.
- [17] Ana Lopes. *Speeding up machine learning for particle physics*. Jun. de 2021. URL: <https://home.cern/news/news/physics/speeding-machine-learning-particle-physics>.
- [18] Aitor Sánchez Mansilla. *Nuevos avances en tomografía cuántica*. Universidad de Zaragoza.
- [19] M. Raissi, P. Perdikaris y G.E. Karniadakis. «Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations». En: *Journal of Computational Physics* 378 (2019), págs. 686-707. DOI: 10.1016/j.jcp.2018.10.045.
- [20] Jun John Sakurai. *Modern quantum mechanics*. Reading, MA: Addison-Wesley, 1994. URL: <https://cds.cern.ch/record/1167961>.
- [21] John A. Smolin, Jay M. Gambetta y Graeme Smith. «Maximum Likelihood, Minimum Effort». En: *Physical Review Letters* 108.7 (feb. de 2012). DOI: 10.1103/physrevlett.108.070502.

- [22] Courtney J. Sporer et al. «Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision». En: *PLOS Computational Biology* 16.10 (oct. de 2020), págs. 1-27. DOI: 10.1371/journal.pcbi.1008215.
- [23] Roberto Tagliaferri et al. «Neural networks in astronomy». En: *Neural Networks* 16.3 (2003). Neural Network Analysis of Complex Scientific Data: Astronomy and Geosciences, págs. 297-319. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(03)00028-5.
- [24] Giacomo Torlai y Roger G. Melko. «Machine-Learning Quantum States in the NISQ Era». En: *Annual Review of Condensed Matter Physics* 11.1 (2020), págs. 325-344. DOI: 10.1146/annurev-conmatphys-031119-050651.
- [25] Julio Íñigo de Vicente Majua. «Medidas de información, incertidumbre y entrelazamiento en Mecánica cuántica». En: (2008).
- [26] Stephen Walborn et al. «Experimental determination of entanglement with a single measurement». En: *Nature* 440 (mayo de 2006), págs. 1022-4. DOI: 10.1038/nature04627.
- [27] *What are Neural Networks?* URL: <https://www.ibm.com/topics/neural-networks> (visitado 14-10-2023).
- [28] Chenguang Yang et al. «Neural Network for Complex Systems: Theory and Applications». En: *Complexity* 2018 (mayo de 2018), págs. 1-2. DOI: 10.1155/2018/3141805.