**Cuestionario final**

**Juan David Abaunza**

**Pregunta 1 (10 puntos) - Fundamentos de CNN**

Explique qué son las redes neuronales convolucionales (CNN) y por qué son especialmente efectivas para el reconocimiento de imágenes. Mencione al menos tres características distintivas de las CNN que las hacen superiores a las redes neuronales tradicionales para tareas de visión por computadora.

**R//**

**¿Qué son?**

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son un tipo especializado de red neuronal diseñada para procesar datos con una estructura en forma de cuadrícula, como las imágenes.

**¿Por qué son efectivas para el reconocimiento de imágenes?**

Las CNN son especialmente efectivas porque extraen automáticamente características relevantes de las imágenes utilizando filtros (o kernels) que detectan patrones locales, como bordes, texturas o formas.

**Tres características distintivas de las CNN:**

**Capas convolucionales (Convolutional Layers):** Aplican filtros sobre pequeñas regiones locales de la imagen para detectar patrones espaciales. Esto permite a la red aprender características relevantes de forma eficiente y reutilizable.

**Compartición de pesos (Weight Sharing):** Un mismo filtro se aplica a toda la imagen, lo que reduce drásticamente el número de parámetros comparado con una red totalmente conectada y mejora la generalización.

**Conectividad local (Local Connectivity):** Cada neurona de una capa convolucional está conectada solo a una pequeña región del mapa de activación anterior, lo que permite capturar características locales sin procesar toda la imagen de una vez.

**Pregunta 2 (10 puntos) - Arquitectura y Componentes**

Describa las capas principales que componen una CNN típica (Conv2D, MaxPooling2D, Flatten, Dense) y explique la función específica de cada una en el proceso de reconocimiento de imágenes. ¿Por qué es importante el orden de estas capas?

**R//**

**Conv2D (Capa convolucional en 2D)**

**Función:** Aplica filtros (kernels) sobre la imagen o mapa de características para detectar patrones locales como bordes, texturas, formas, etc.

**Salida:** Un mapa de activación que resalta las características detectadas por cada filtro.

**Importancia:** Es el núcleo del aprendizaje de características en CNN.

**MaxPooling2D (Capa de agrupamiento máximo en 2D)**

**Función:** Reduce la dimensión espacial (alto y ancho) del mapa de activación seleccionando el valor máximo dentro de una ventana (por ejemplo, 2x2).

**Beneficios:**

* Reduce el número de parámetros y el costo computacional.
* Ayuda a que la red sea más invariante a traslaciones (pequeños desplazamientos en la imagen).
* Reduce el riesgo de sobreajuste.

**Flatten (Aplanamiento)**

**Función:** Convierte el mapa de activación multidimensional (por ejemplo, 7x7x64) en un vector unidimensional para conectarse con capas densas.

**Rol:** Transforma los datos para que puedan ser procesados por capas totalmente conectadas (Dense).

**Dense (Capa totalmente conectada)**

**Función:** Cada neurona está conectada a todas las salidas de la capa anterior. Se encarga de tomar las características extraídas y combinarlas para hacer la predicción final.

**Usos comunes:** Clasificación de imágenes en categorías usando funciones de activación como softmax (para clasificación multiclase).

**¿Por qué es importante el orden de estas capas?**

El orden es crucial porque cada capa realiza una transformación específica y dependiente de la salida de la capa anterior:

* Conv2D primero extrae características.
* MaxPooling2D reduce la dimensionalidad y conserva las más importantes.
* Flatten prepara los datos para una estructura lineal.
* Dense realiza la toma de decisiones basada en las características aprendidas.

**Pregunta 3 (12 puntos) - Preprocesamiento de Datos**

En el contexto del dataset CIFAR-10: a) ¿Por qué es necesario normalizar los valores de píxeles al rango [0, 1]? (4 puntos) b) ¿Qué significa convertir las etiquetas a formato "one-hot" y por qué es necesario? (4 puntos) c) Mencione dos técnicas de data augmentation que podrían mejorar el rendimiento del modelo y explique cómo funcionan. (4 puntos)

**R//**

El preprocesamiento es una etapa crítica antes de entrenar redes neuronales, ya que mejora la estabilidad del entrenamiento, acelera la convergencia y ayuda al modelo a generalizar mejor. A continuación, se responde cada parte en el contexto del dataset CIFAR-10, que contiene 60,000 imágenes de 10 clases (como aviones, autos, gatos, etc.).

**¿Por qué es necesario normalizar los valores de píxeles al rango [0, 1]?**

Las imágenes del CIFAR-10 tienen píxeles con valores en el rango [0, 255]. Normalizar esos valores al rango [0, 1] (dividiendo entre 255) es importante porque:

* Mejora la estabilidad numérica durante el entrenamiento.
* Acelera la convergencia, ya que los valores de entrada están en un rango pequeño y uniforme.
* Las redes neuronales funcionan mejor cuando los datos de entrada tienen una escala similar, especialmente si se usan funciones de activación como ReLU o sigmoide.

**¿Qué significa convertir las etiquetas a formato "one-hot" y por qué es necesario?**

Convertir a formato "one-hot" significa representar cada clase como un vector binario en el que solo una posición es 1 y el resto son 0. Por ejemplo, si una imagen pertenece a la clase 3 de un total de 10, su representación será:

[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

**Razones para usar one-hot encoding:**

* Es necesario para usar funciones de pérdida como categorical crossentropy, que requieren vectores de probabilidad como etiquetas.
* Permite al modelo aprender a asignar una probabilidad a cada clase.
* Evita que la red interprete relaciones ordinales entre clases (por ejemplo, pensar que la clase 2 es "más" que la clase 1), lo que sí pasaría si las etiquetas fueran números enteros.

**dos técnicas de data augmentation que podrían mejorar el rendimiento del modelo y explique cómo funcionan**

**Rotación aleatoria (Random Rotation):**

Aplica rotaciones pequeñas a las imágenes (por ejemplo, ±15°).

**Función:** Ayuda al modelo a ser invariante a la orientación del objeto, mejorando su capacidad de generalizar.

**Desplazamiento horizontal (Horizontal Flip):**

Invierte la imagen horizontalmente con cierta probabilidad (por ejemplo, 50%).

**Función:** Simula que un objeto puede estar orientado hacia la izquierda o derecha, útil para clases como "automóvil" o "pájaro".

**Pregunta 4 (10 puntos) - Optimización y Entrenamiento**

Analice los siguientes aspectos del entrenamiento de una CNN:

* ¿Qué función de pérdida se utiliza para clasificación multiclase y por qué?
* ¿Cuál es la diferencia entre usar 'adam' y 'sgd' como optimizadores?
* ¿Cómo se puede detectar y prevenir el overfitting durante el entrenamiento?

**R//**

**¿Qué función de pérdida se utiliza para clasificación multiclase y por qué?**

Se utiliza la función de pérdida categorical crossentropy (entropía cruzada categórica).

**Razón:**

* Es adecuada para tareas de clasificación multiclase donde cada ejemplo pertenece a una sola clase.
* Compara la distribución de probabilidades predicha (salida de softmax) con la verdadera (vector one-hot).
* Penaliza más fuertemente las predicciones erróneas con alta confianza, lo que guía mejor al modelo a ajustar sus predicciones.

**¿Cuál es la diferencia entre usar 'adam' y 'sgd' como optimizadores?**

|  | **SGD (Stochastic Gradient Descent)** | **Adam (Adaptive Moment Estimation)** |
| --- | --- | --- |
| **Velocidad de convergencia** | Más lenta, sensible al learning rate | Más rápida, adapta el learning rate |
| **Estabilidad** | Puede estancarse si no está bien ajustado | Más robusto a diferentes configuraciones |
| **Adaptación** | Usa una tasa de aprendizaje fija (puede decaer) | Ajusta automáticamente la tasa de aprendizaje por parámetro |
| **Uso típico** | Preferido para modelos grandes con ajuste fino | Ideal para prototipos y entrenamientos iniciales |

**¿Cómo se puede detectar y prevenir el overfitting durante el entrenamiento?**

**Detección:**

* Se observa cuando la pérdida de entrenamiento sigue disminuyendo mientras que la pérdida de validación empieza a aumentar.
* También puede notarse si hay una gran diferencia entre la precisión en entrenamiento y validación.

**Prevención (técnicas comunes):**

**Regularización (L2 o Dropout):**

* Dropout desactiva aleatoriamente algunas neuronas durante el entrenamiento, evitando la dependencia excesiva de ciertas rutas de activación.

**Data Augmentation:**

* Aumenta artificialmente el tamaño y variedad del conjunto de entrenamiento, haciendo que el modelo generalice mejor.

**Early Stopping:**

* Detiene el entrenamiento cuando la pérdida de validación deja de mejorar después de varios intentos, evitando el sobreentrenamiento.

**Reducir la complejidad del modelo:**

* Usar menos capas o menos neuronas si el modelo es demasiado grande para el conjunto de datos disponible.

**Pregunta 5 (10 puntos) - Transfer Learning**

El taller menciona el uso de MobileNetV2 pre-entrenado. Explique:

* ¿Qué es transfer learning y cuáles son sus ventajas?
* ¿Por qué se eligió MobileNetV2 para este proyecto específico?
* ¿En qué situaciones sería preferible entrenar un modelo desde cero versus usar transfer learning?

**R//**

**¿Qué es transfer learning y cuáles son sus ventajas?**

Transfer learning consiste en utilizar un modelo previamente entrenado (por ejemplo, en un gran dataset como ImageNet) y adaptarlo a una nueva tarea (como clasificar imágenes de CIFAR-10 o un conjunto personalizado).

**Ventajas:**

**Ahorro de tiempo de entrenamiento:** No se parte desde cero.

**Mejor rendimiento con menos datos:** Útil cuando el nuevo dataset es pequeño o limitado.

**Aprovecha conocimientos previos:** Las primeras capas de un modelo preentrenado ya han aprendido a detectar patrones generales como bordes, formas y texturas, útiles para muchas tareas visuales.

**¿Por qué se eligió MobileNetV2 para este proyecto específico?**

MobileNetV2 es una arquitectura ligera y eficiente que se diseñó para funcionar bien en dispositivos con recursos limitados (como móviles o sistemas embebidos).

**Razones para elegirla:**

**Modelo compacto:** Tiene menos parámetros que modelos grandes como ResNet o VGG, lo que permite entrenar y ejecutar más rápido.

**Buena precisión:** A pesar de su tamaño, mantiene un rendimiento competitivo en muchas tareas de clasificación.

**Ideal para prototipos:** prototipos o aplicaciones en tiempo real, donde el rendimiento y la velocidad son importantes.

**¿En qué situaciones sería preferible entrenar un modelo desde cero versus usar transfer learning?**

**Cuando se dispone de un gran conjunto de datos propio:**

Si tienes millones de ejemplos bien etiquetados, un modelo personalizado desde cero puede aprender patrones específicos sin depender de otros dominios.

**Cuando la tarea o los datos son muy diferentes del modelo preentrenado:**

Si los datos son muy distintos a los del dataset original (por ejemplo, imágenes médicas en lugar de imágenes naturales), las características preaprendidas pueden no ser útiles, y entrenar desde cero puede ser más efectivo.

**Pregunta 6 (12 puntos) - Procesamiento de Lenguaje Natural**

Respecto al componente NLP del sistema integrado: a) Explique qué es la lemmatización y por qué es importante en el procesamiento de texto. (4 puntos) b) ¿Cómo funcionan los patrones de conversación definidos en el código para identificar intenciones del usuario? (4 puntos) c) Mencione tres técnicas que podrían mejorar la capacidad de comprensión del chatbot. (4 puntos)

**R//**

**¿Qué es la lematización y por qué es importante en el procesamiento de texto?**

Es el proceso de reducir una palabra a su forma base o "lema", que es su forma canónica. A diferencia del stemming, que simplemente corta sufijos, la lematización utiliza reglas lingüísticas para transformar las palabras a su raíz correcta.

**Ejemplo:** “corriendo”, “corre”, “corrí” → “correr”

**Importancia:**

* Permite unificar palabras con el mismo significado, mejorando la comprensión semántica.
* Reduce la dimensionalidad del texto, lo que facilita el análisis y mejora la precisión del modelo de NLP.
* Ayuda a identificar intenciones del usuario con más claridad, al agrupar variaciones de una misma acción o concepto.

**¿Cómo funcionan los patrones de conversación definidos en el código para identificar intenciones del usuario?**

En sistemas básicos de NLP como los desarrollados en talleres, se definen patrones de texto para cada intención (intents) en forma de expresiones comunes que los usuarios podrían usar.

**Funcionamiento:**

* Cada intención tiene un conjunto de frases o patrones (patterns) posibles (por ejemplo: “¿cómo estás?”, “hola”, “buenos días”).
* Se usa preprocesamiento (tokenización, lematización, etc.) para convertir esas frases a un formato numérico.
* Un modelo de clasificación (como una red neuronal sencilla) compara la entrada del usuario con estos patrones.
* El modelo predice la intención (por ejemplo, "saludo", "pregunta por horarios", etc.) y genera una respuesta asociada.

**tres técnicas que podrían mejorar la capacidad de comprensión del chatbot**

**Embeddings semánticos (como Word2Vec, GloVe o BERT):**

Representan palabras en un espacio vectorial que captura relaciones semánticas y contextuales, permitiendo al chatbot comprender mejor el significado real del texto.

**Modelos de lenguaje avanzados (como BERT, GPT o transformers):**

Permiten entender el contexto completo de la oración, incluso en preguntas largas o ambiguas, mejorando la precisión en la detección de intenciones.

**Reconocimiento de entidades nombradas (NER - Named Entity Recognition):**

Permite identificar elementos clave en el texto, como nombres de personas, lugares, fechas, etc., lo que ayuda al chatbot a entender detalles específicos del mensaje del usuario.

**Pregunta 7 (10 puntos) - Integración de Sistemas**

El taller propone integrar reconocimiento de imágenes con NLP. Describa:

* ¿Cuáles son los principales desafíos técnicos de esta integración?
* ¿Cómo se mantiene el contexto entre el análisis de imágenes y la conversación?
* Proponga una mejora específica para hacer más fluida esta integración.

**R//**

**¿Cuáles son los principales desafíos técnicos de esta integración?**

**Comunicación entre módulos heterogéneos:**

* El módulo de visión por computadora y el módulo NLP están basados en tecnologías diferentes (por ejemplo, TensorFlow para imágenes y NLTK o spaCy para NLP).
* Requieren sincronización y formatos de datos compatibles.

**Mantenimiento del flujo lógico de la interacción:**

* Es difícil coordinar el momento en que se analiza una imagen con la generación de una respuesta verbal relevante.
* Hay riesgo de que el chatbot pierda el hilo conversacional si no se gestiona el estado adecuadamente.

**Consumo de recursos y rendimiento:**

* Procesar imágenes y texto al mismo tiempo puede afectar el rendimiento, especialmente en dispositivos con recursos limitados.

**¿Cómo se mantiene el contexto entre el análisis de imágenes y la conversación?**

Se puede mantener el contexto utilizando un sistema de gestión de estado (state management), como un diccionario o estructura de datos que almacene la información relevante del usuario durante la sesión

**Ejemplo:**

Si el usuario envía una imagen de un gato, el sistema guarda:

* última\_imagen = "gato"
* intención\_actual = "consultar\_animal"

**Con esta información, el chatbot puede responder de forma coherente:**

* Usuario: “¿Qué animal es este?”
* Bot (usa última\_imagen): “Parece ser un gato doméstico.”

**Mejora específica para hacer más fluida esta integración**

**Mejora propuesta:**

Implementar un gestor de contexto conversacional con memoria a corto plazo, usando herramientas como Rasa o transformers con capacidades de diálogo contextual (ej. DialoGPT, ChatGPT API).

**Beneficios:**

* Permite al sistema recordar interacciones anteriores y adaptar sus respuestas.
* Facilita una conversación dinámica, donde el chatbot puede hacer referencia a análisis de imágenes anteriores sin perder coherencia.
* Puede integrar múltiples fuentes de entrada (texto, imagen) y generar respuestas personalizadas y más naturales.

**Pregunta 8 (8 puntos) - Análisis de Rendimiento**

Para evaluar el desempeño de una CNN: a) ¿Qué información proporciona una matriz de confusión? (4 puntos) b) ¿Cuál es la diferencia entre accuracy, precision y recall? ¿Cuándo es más importante cada métrica? (4 puntos)

**R//**

**¿Qué información proporciona una matriz de confusión?**

La matriz de confusión es una tabla que resume el desempeño de un modelo de clasificación, comparando las etiquetas verdaderas con las predicciones del modelo.

**¿Qué muestra?**

* Las filas representan las clases reales (etiquetas verdaderas).
* Las columnas representan las clases predichas por el modelo.

**¿Qué información útil proporciona?**

**Aciertos (diagonal):** Número de predicciones correctas por clase.

**Errores (fuera de la diagonal):** Muestra qué clases se están confundiendo entre sí.

**Clases difíciles:** Permite detectar clases difíciles para el modelo y desequilibrios de desempeño.

**¿Cuál es la diferencia entre accuracy, precision y recall? ¿Cuándo es más importante cada métrica?**

| **Métrica** | **Fórmula (Binaria)** | **¿Qué mide?** |
| --- | --- | --- |
| **Accuracy** | (TP + TN) / (TP + TN + FP + FN) | Porcentaje total de predicciones correctas. |
| **Precision** | TP / (TP + FP) | Qué proporción de las predicciones positivas fueron correctas. |
| **Recall** (sensibilidad) | TP / (TP + FN) | Qué proporción de los casos positivos reales fueron detectados. |

**Dónde cada una es más importante:**

**Accuracy:**

* Útil cuando las clases están equilibradas.
* No confiable si hay desbalance (por ejemplo, 95% de una clase y 5% de otra).

**Precision:**

* Importante cuando los falsos positivos son costosos.
* Ejemplo: Clasificación de correos spam (no queremos etiquetar correos válidos como spam).

**Recall:**

* Clave cuando los falsos negativos son críticos.
* Ejemplo: Diagnóstico médico (mejor detectar todos los casos enfermos, aunque haya algunos falsos positivos).

**Pregunta 9 (10 puntos) - Casos de Uso Específicos**

El sistema se plantea para apoyo académico y consultas psicológicas básicas. Analice:

* ¿Qué consideraciones éticas y de privacidad se deben tener en cuenta?
* ¿Cómo se podría adaptar el sistema para detectar situaciones que requieran intervención humana?
* Proponga una funcionalidad adicional que agregue valor al sistema.

**R//**

**¿Qué consideraciones éticas y de privacidad se deben tener en cuenta?**

**Privacidad de los datos personales:**

* El sistema puede recibir información sensible (como estados emocionales o problemas personales), por lo que debe garantizar confidencialidad.
* Se deben aplicar políticas de protección de datos (como el GDPR o leyes locales) y asegurar el uso de canales cifrados para las comunicaciones.

**Consentimiento informado:**

* Los usuarios deben ser informados sobre cómo se usan sus datos, qué hace el sistema, y que no reemplaza a un profesional de salud mental.

**Limitaciones del sistema:**

* El chatbot debe dejar claro que ofrece orientación básica y no puede diagnosticar ni tratar condiciones psicológicas graves.

**Evitar sesgos y discriminación:**

* Asegurar que los modelos no reproduzcan estereotipos o sesgos en sus respuestas, especialmente en contextos emocionales o académicos.

**¿Cómo se podría adaptar el sistema para detectar situaciones que requieran intervención humana?**

El sistema puede incluir un módulo de detección de alertas críticas, basado en NLP, que identifique frases relacionadas con:

**Riesgo emocional o crisis:** Ej. “ya no quiero seguir”, “me siento inútil”, “nadie me entiende”.

**Casos de abuso o acoso:** Ej. “me están amenazando”, “me siento acosado/a”.

**Implementación:**

Crear una lista de palabras clave y patrones emocionales críticos.

Si se detecta una alerta:

* El sistema puede sugerir contactar a un profesional.
* Activar una notificación a un supervisor humano (con el consentimiento del usuario).
* Ofrecer recursos como líneas de ayuda o contacto con psicólogos institucionales.

**funcionalidad adicional que agregue valor al sistema**

**Funcionalidad propuesta:** “Bitácora de Bienestar Personal”

**¿Qué es?**

Un módulo donde el usuario pueda registrar cómo se siente a diario, responder breves preguntas de autocuidado o estado emocional.

**Beneficios:**

* Permite al sistema detectar patrones emocionales a lo largo del tiempo.
* El usuario puede ver su progreso, lo que fomenta el autorreflexión y el autocuidado.
* Puede generar informes privados útiles en caso de hablar con un orientador o profesional.

**Pregunta 10 (8 puntos) - Visión Futura**

Considerando las extensiones propuestas en el taller:

* ¿Cuál de las extensiones mencionadas considera más importante implementar primero y por qué?
* ¿Cómo impactarían los avances en modelos de lenguaje como GPT o BERT en este tipo de sistemas?

**R//**

**¿Cuál de las extensiones mencionadas considera más importante implementar primero y por qué?**

**Extensión prioritaria:**

Detección de emociones o estados de ánimo del usuario a partir del lenguaje.

**¿Por qué es la más importante?**

* Permite personalizar las respuestas del chatbot según el estado emocional del usuario, especialmente relevante en contextos de apoyo psicológico.
* Aumenta la empatía percibida, mejorando la experiencia y confianza del usuario.
* Ayuda a detectar casos sensibles o de riesgo emocional con más precisión, incluso si el usuario no es explícito.

**Ejemplo de aplicación:**

Si el usuario dice “No tengo ganas de estudiar hoy”, el sistema puede detectar un posible estado de ánimo bajo y ofrecer palabras de ánimo o sugerir recursos de bienestar.

**¿Cómo impactarían los avances en modelos de lenguaje como GPT o BERT en este tipo de sistemas?**

**Mejor comprensión del lenguaje natural:**

Modelos como GPT y BERT comprenden el contexto semántico completo de una oración, lo que mejora la detección de intenciones, incluso si se expresan de forma ambigua, informal o emocional.

**Generación de respuestas más naturales y empáticas:**

GPT puede generar respuestas más humanas, contextuales y coherentes, permitiendo conversaciones más fluidas y naturales.

**Mejor manejo del contexto conversacional:**

Los transformers pueden recordar interacciones anteriores y adaptar las respuestas en consecuencia, lo que mejora la continuidad y la personalización del diálogo.

**Adaptabilidad a múltiples tareas:**

Un solo modelo puede manejar múltiples funciones: clasificación de intenciones, análisis emocional, generación de respuestas, etc., reduciendo la complejidad del sistema.