

Modelo de Reconocimiento Automático de Señales de Tránsito Vehicular mediante Aprendizaje Profundo de Redes Neuronales Convolucionales

Josué Gastón Távara Idrogo jtavara@unitru.edu.pe

Universidad Nacional de Trujillo Facultad de Ciencias Físicas y Matemáticas Escuela Académico
Profesional de Informática

Defensa de tesis 15/12/2018



Resumen

Esta investigación pretende contribuir en la industria automotriz, específicamente en los campos de construcción de vehículos autónomos y de los sistemas avanzados de asistencia al conductor, iniciando con la adquisición de imágenes, luego para el pre-procesamiento, se implementará algoritmos de realce de contraste, reducción de ruido, rotación y proyecciones de escalamiento con la finalidad de aumentar el conjunto de datos y poder ejecutar el aprendizaje profundo a través de arquitecturas de redes neuronales convolucionales. Se realizaron diferentes diseños de arquitecturas convolucionales y se escogió el que obtuvo los mejores indicadores/resultados.

Palabras claves: aprendizaje profundo, redes neuronales convolucionales, procesamiento de imágenes.

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

1 Introducción

2 Motivación

3 Formulación del problema

4 Importancia de la investigación

5 Contribución de la investigación

6 Marco teórico

- Aprendizaje Profundo
- Red Convolutacional
- Arquitectura del Modelo
- Componentes del Modelo
- Validación Cruzada

7 Propuesta o tema central de la tesis

8 Resultados de la tesis

9 Consideraciones finales

10 Referencias bibliograficas

Introducción

- Al conducir en carreteras congestionadas, a veces es difícil mantener los ojos en todas partes a la vez, comprobando el camino por delante, el tráfico venidero, lo que está detrás de usted, tratar de mantener la velocidad permitida; es por ello, que existen mecanismos destinados a reglamentar el tránsito, advertir o informar a los usuarios mediante palabras, sonidos o símbolos determinados.

Introducción

- Al conducir en carreteras congestionadas, a veces es difícil mantener los ojos en todas partes a la vez, comprobando el camino por delante, el tráfico venidero, lo que está detrás de usted, tratar de mantener la velocidad permitida; es por ello, que existen mecanismos destinados a reglamentar el tránsito, advertir o informar a los usuarios mediante palabras, sonidos o símbolos determinados.
- La policía de tránsito o las **señalizaciones vehiculares** que según sea el caso, en todos los países regulan el tránsito e informan al usuario sobre direcciones, rutas, destinos, así como dificultades existentes en las carreteras y previenen cualquier peligro que podría presentarse en la circulación vehicular.

Introducción

- La inseguridad vial es un problema de interés mundial, según el último informe de la OMS (Organización Mundial de la salud) anualmente cerca de 1,3 millones de personas mueren alrededor del mundo y entre 20 y 50 millones padecen traumatismos no mortales,(OMS, 2017). Son distintas las causas que llevan a este problema, de las cuales las principales pueden ser la falta de concientización y educación vial.

Introducción

- El Perú y la capital Lima se encuentran respectivamente en la lista de peores países y ciudades para conducir en America Latina, según el Índice Global de Satisfacción del Conductor (Waze, 2016), lo cual se ve reflejado en que los últimos años se ha incrementado el índice de mortandad originados por los accidentes de tránsito siendo las principales causas de los mismos el exceso de velocidad, estado de ebriedad del conductor y sobretodo el desacato a las señales de tránsito, todas ellas de responsabilidad directa del conductor del vehículo motorizado,(SUTRAN, 2014).

Introducción

- El Perú y la capital Lima se encuentran respectivamente en la lista de peores países y ciudades para conducir en America Latina, según el Índice Global de Satisfacción del Conductor (Waze, 2016), lo cual se ve reflejado en que los últimos años se ha incrementado el índice de mortandad originados por los accidentes de tránsito siendo las principales causas de los mismos el exceso de velocidad, estado de ebriedad del conductor y sobretodo el desacato a las señales de tránsito, todas ellas de responsabilidad directa del conductor del vehículo motorizado,(SUTRAN, 2014).
- Es por ello que trabajar en obtener vehículos más seguros es un factor fundamental para prevenir de alguna forma los accidentes de tránsito o reducir la probabilidad de que estos sean producidos, (OMS, 2017).

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

1 Introducción

2 Motivación

3 Formulación del problema

4 Importancia de la investigación

5 Contribución de la investigación

6 Marco teórico

- Aprendizaje Profundo
- Red Convolutacional
- Arquitectura del Modelo
- Componentes del Modelo
- Validación Cruzada

7 Propuesta o tema central de la tesis

8 Resultados de la tesis

9 Consideraciones finales

10 Referencias bibliograficas

Motivación

- Para contribuir con lo antes mencionado, se han venido planteando formas que permitan la asistencia en el reconocimiento de señales de tránsito, la cual es un problema de clasificación multicategórica que comúnmente presenta desigualdades en las frecuencias de aparición de las categorías. Además, las señales de tránsito muestran una amplia gama de variaciones entre las clases en términos de color, forma, subconjuntos de clases que son muy similares entre sí y la presencia de símbolos, leyendas o texto. A esto es sumado, las grandes variaciones en las apariencias visuales debido a cambios de iluminación, occlusiones parciales, rotaciones, condiciones meteorológicas, escalamiento, etc.

Motivación

- Para contribuir con lo antes mencionado, se han venido planteando formas que permitan la asistencia en el reconocimiento de señales de tránsito, la cual es un problema de clasificación multicategórica que comúnmente presenta desigualdades en las frecuencias de aparición de las categorías. Además, las señales de tránsito muestran una amplia gama de variaciones entre las clases en términos de color, forma, subconjuntos de clases que son muy similares entre sí y la presencia de símbolos, leyendas o texto. A esto es sumado, las grandes variaciones en las apariencias visuales debido a cambios de iluminación, occlusiones parciales, rotaciones, condiciones meteorológicas, escalamiento, etc.
- Todo esto representa un reto para el reconocedor/clasificador de señales de tránsito vehicular y es por ello que se han venido realizando diversas investigaciones, donde esta forma parte de una de ellas.

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 **Formulación del problema**
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Formulación del problema

En este trabajo, se propone discutir el modelo de redes neuronales convolucionales basado en el problema del reconocimiento de imágenes para responder a la siguiente pregunta:

¿Cómo se puede reconocer de manera automática señales de tránsito vehicular?

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Importancia de la investigación - Justificación Académica

- La importancia de esta investigación en el punto de vista de ciencias de la computación se justifica en poner en práctica los conocimientos adquiridos en la formación académica, siendo los más resaltantes el tema de procesamiento de imágenes, cálculo matemático e inteligencia artificial con la finalidad de obtener un modelo robusto de redes neuronales convolucionales basadas en el aprendizaje profundo (deep learning) que permita analizar el contenido de imágenes para el reconocer de señales de tránsito vehicular.

Importancia de la investigación - Justificación Académica

- La importancia de esta investigación en el punto de vista de ciencias de la computación se justifica en poner en práctica los conocimientos adquiridos en la formación académica, siendo los más resaltantes el tema de procesamiento de imágenes, cálculo matemático e inteligencia artificial con la finalidad de obtener un modelo robusto de redes neuronales convolucionales basadas en el aprendizaje profundo (deep learning) que permita analizar el contenido de imágenes para el reconocer de señales de tránsito vehicular.
- En este sentido, esta investigación pretende la elaboración de un modelo basado en el aprendizaje profundo de redes convolucionales que permita el reconocimiento de señales de tránsito vehicular. La siguiente figura muestra un diagrama de bloques con la secuencia de actividades que demanda el reconocimiento.

Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito en la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y muertes se puedan evitar y en consecuencia reducir estos índices progresivamente.

Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito en la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y muertes se puedan evitar y en consecuencia reducir estos índices progresivamente.
- A continuación algunos ejemplos:

Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito en la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y muertes se puedan evitar y en consecuencia reducir estos índices progresivamente.
- A continuación algunos ejemplos:
 - Dar una notificación al no darse cuenta de un cambio en el límite de velocidad

Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito en la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y muertes se puedan evitar y en consecuencia reducir estos índices progresivamente.
- A continuación algunos ejemplos:
 - Dar una notificación al no darse cuenta de un cambio en el límite de velocidad
 - El aviso de que se está cometiendo una infracción al girar o estacionarse donde no se debe

Importancia de la investigación - Justificación Social

Importancia de la investigación - Justificación Social

- — La advertencia de un peligro potencial por delante.

Importancia de la investigación - Justificación Social

- - La advertencia de un peligro potencial por delante.
 - Una aplicación móvil que dé la posibilidad de reconocer automáticamente aquella señal de tránsito que usuarios puedan desconocer serviría como aporte en la educación vial.

Importancia de la investigación - Justificación Social

- - La advertencia de un peligro potencial por delante.
 - Una aplicación móvil que dé la posibilidad de reconocer automáticamente aquella señal de tránsito que usuarios puedan desconocer serviría como aporte en la educación vial.
 - Podría ser un componente útil en cámaras de video vigilancia convirtiéndolas en más inteligentes con la capacidad de monitorear que las señales de tránsito sean respetadas.

Importancia de la investigación - Justificación Social

- - La advertencia de un peligro potencial por delante.
 - Una aplicación móvil que dé la posibilidad de reconocer automáticamente aquella señal de tránsito que usuarios puedan desconocer serviría como aporte en la educación vial.
 - Podría ser un componente útil en cámaras de video vigilancia convirtiéndolas en más inteligentes con la capacidad de monitorear que las señales de tránsito sean respetadas.
- En este sentido, esta investigación pretende la elaboración de un modelo basado en el aprendizaje profundo de redes convolucionales que permita el reconocimiento de señales de tránsito vehicular. La siguiente figura muestra un diagrama de bloques con la secuencia de actividades que demanda el reconocimiento.

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Contribución de la investigación

- Todos los hechos descritos hacen que el reconocimiento de las señales de tránsito sea un reto desafiante y esencial en muchos aspectos, no solo para contribuir en los esfuerzos de la industria automotriz en el campo de la asistencia al conductor, sino también para organismos internacionales y gubernamentales quienes se dan cuenta de la problemática que representa la inseguridad vial y buscan constantemente introducir nuevos mecanismos y tecnologías que faciliten y mejoren la conducción vehicular para el beneficio propio del conductor y en general para la seguridad vial de la sociedad.

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Aprendizaje Profundo

- Aprendizaje Profundo

Aprendizaje Profundo

● Aprendizaje Profundo

- ▶ El verdadero desafío para la inteligencia artificial fue y es resolver las tareas que son fáciles de realizar para las personas pero difíciles de describir de manera formal, problemas que resolvemos intuitivamente, que se sienten automáticos, como reconocer palabras, rostros u objetos en las imágenes (Goodfellow et al., 2016).

Aprendizaje Profundo

● Aprendizaje Profundo

- ▶ El verdadero desafío para la inteligencia artificial fue y es resolver las tareas que son fáciles de realizar para las personas pero difíciles de describir de manera formal, problemas que resolvemos intuitivamente, que se sienten automáticos, como reconocer palabras, rostros u objetos en las imágenes (Goodfellow et al., 2016).
- ▶ El aprendizaje automático es considerada como la mejor técnica de la inteligencia artificial(IA); es decir, es el campo de la IA que hoy en día muestra la mayor promesa al proporcionar herramientas que la industria y la sociedad pueden usar para producir algún cambio.

Aprendizaje Profundo

● Aprendizaje Profundo

- ▶ El verdadero desafío para la inteligencia artificial fue y es resolver las tareas que son fáciles de realizar para las personas pero difíciles de describir de manera formal, problemas que resolvemos intuitivamente, que se sienten automáticos, como reconocer palabras, rostros u objetos en las imágenes (Goodfellow et al., 2016).
- ▶ El aprendizaje automático es considerada como la mejor técnica de la inteligencia artificial(IA); es decir, es el campo de la IA que hoy en día muestra la mayor promesa al proporcionar herramientas que la industria y la sociedad pueden usar para producir algún cambio.
- ▶ En tal sentido,el aprendizaje automático toma algunas de las ideas centrales de la inteligencia artificial y las enfoca en resolver problemas del mundo real con redes neuronales diseñadas para imitar nuestra propia toma de decisiones.

Aprendizaje Profundo

- Por otro lado, el aprendizaje profundo o avanzado de máquinas (del inglés, **deep learning**) se centra aún más estrechamente en un subconjunto de herramientas y técnicas del aprendizaje automático y los aplica a la solución de casi cualquier problema que requiera "pensamiento", ya sea humano o artificial.

Aprendizaje Profundo

- Por otro lado, el aprendizaje profundo o avanzado de máquinas (del inglés, **deep learning**) se centra aún más estrechamente en un subconjunto de herramientas y técnicas del aprendizaje automático y los aplica a la solución de casi cualquier problema que requiera "pensamiento", ya sea humano o artificial.
- Se puede sostener que el aprendizaje automático es el único enfoque viable para construir sistemas de inteligencia artificial que puedan operar en entornos complicados del mundo real y el **aprendizaje profundo** es, a su vez, un tipo particular de aprendizaje automático que logra gran poder y flexibilidad al representar el mundo como una jerarquía de conceptos anidados, con cada concepto definido en relación con conceptos más simples y representaciones más abstractas calculadas en base a representaciones menos abstractas.

Aprendizaje Profundo

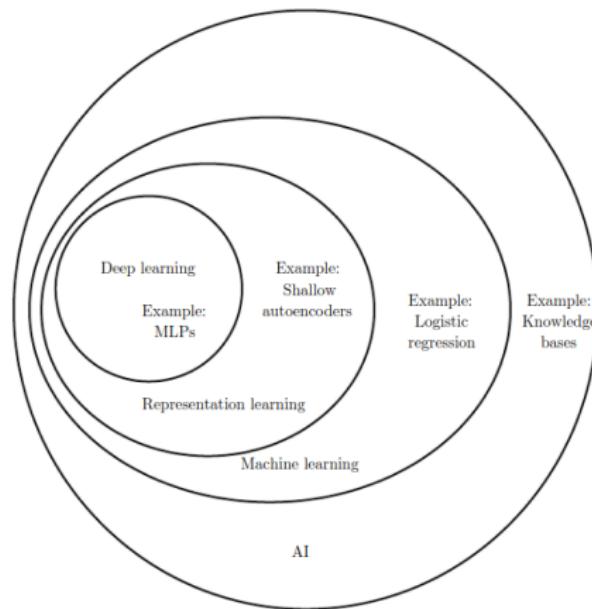


Figura: Diagrama de Venn donde cada sección incluye un ejemplo de una tecnología de IA (ilustra la relación entre estas diferentes disciplinas de la IA)

Fuente: (Goodfellow et al., 2016)

Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).

Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).
- Se considera que un algoritmo es profundo si los datos de entrada se pasan a través de una serie de no linealidades o transformaciones no lineales antes de que se emita.

Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).
- Se considera que un algoritmo es profundo si los datos de entrada se pasan a través de una serie de no linealidades o transformaciones no lineales antes de que se emita.
- Este enfoque permite que las computadoras aprendan de la experiencia y entiendan el mundo en términos de una jerarquía de conceptos, con cada concepto definido a través de su relación con conceptos más simples.

Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).
- Se considera que un algoritmo es profundo si los datos de entrada se pasan a través de una serie de no linealidades o transformaciones no lineales antes de que se emita.
- Este enfoque permite que las computadoras aprendan de la experiencia y entiendan el mundo en términos de una jerarquía de conceptos, con cada concepto definido a través de su relación con conceptos más simples.
- Al reunir el conocimiento de la experiencia, este enfoque evita la necesidad de que los operadores humanos especifiquen formalmente todo el conocimiento que necesita la computadora.

Aprendizaje Profundo

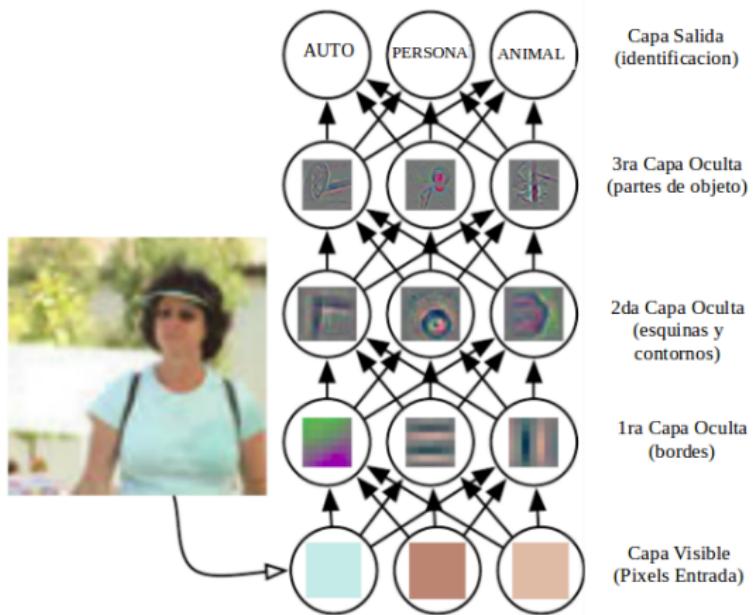


Figura: El aprendizaje profundo permite a la computadora construir conceptos complejos a partir de conceptos más simples

Fuente propia

Red Convolucional

- Nueve de cada diez veces, cuando se escucha que el aprendizaje profundo rompe una nueva barrera tecnológica, las redes neuronales convolucionales están involucradas. También llamados CNN(del inglés , **Convolutional Neural Networks**) o **ConvNets**, estas son las preferidas del campo de redes neuronales profundas. Han aprendido a clasificar las imágenes en categorías incluso mejor que los humanos en algunos casos, (Rohrer, 2016).

Red Convolucional

- Nueve de cada diez veces, cuando se escucha que el aprendizaje profundo rompe una nueva barrera tecnológica, las redes neuronales convolucionales están involucradas. También llamados CNN(del inglés , **Convolutional Neural Networks**) o **ConvNets**, estas son las preferidas del campo de redes neuronales profundas. Han aprendido a clasificar las imágenes en categorías incluso mejor que los humanos en algunos casos, (Rohrer, 2016).
- Estas son muy similares a las redes neuronales ordinarias, están formadas por neuronas que tienen pesos y biases(sesgos) que se pueden aprender durante el entrenamiento de estas. Cada neurona recibe algunas entradas y realiza operaciones matemáticas.

Red Convolucional

- Una ConvNet se caracteriza por tener una secuencia de capas donde cada una de estas transforma un volumen de activaciones en otro nuevo a través de funciones y el aprendizaje profundo se produce cuando se utilizan varias de estas capas variando los parámetros de configuración dentro y entre dichas capas.

Red Convolucional

- Una ConvNet se caracteriza por tener una secuencia de capas donde cada una de estas transforma un volumen de activaciones en otro nuevo a través de funciones y el aprendizaje profundo se produce cuando se utilizan varias de estas capas variando los parámetros de configuración dentro y entre dichas capas.
- El resultado de las CNNs es que pueden encontrar si una característica está en una imagen sin preocuparse exactamente de donde está. Esto ayuda a resolver el problema de las computadoras al comparar imágenes de manera hiper-literal, es decir, que coincida pixel a pixel para que se trate de imágenes iguales.

Red Convolucional

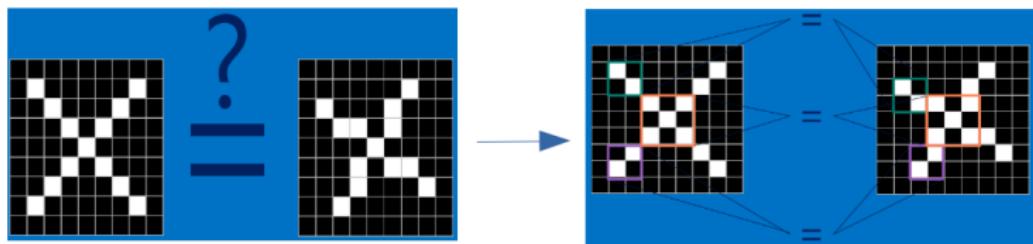


Figura: Análisis de una CNN

Fuente: (Rohrer, 2016)

Red Convolucional

- Red Convolucional - Terminologías

Red Convolucional

- **Red Convolucional - Terminologías**

- ▶ Una es cuando la red convolucional es vista como un número largo de capas simples y cada paso del procesamiento se considera como una capa en sí misma.

Red Convolucional

● Red Convolucional - Terminologías

- ▶ Una es cuando la red convolucional es vista como un número largo de capas simples y cada paso del procesamiento se considera como una capa en sí misma.
- ▶ Otra terminología es cuando la red convolucional es vista como un número pequeño de capas relativamente complejas, donde cada capa tiene múltiples etapas.

Red Convolucional

● Red Convolucional - Terminologías

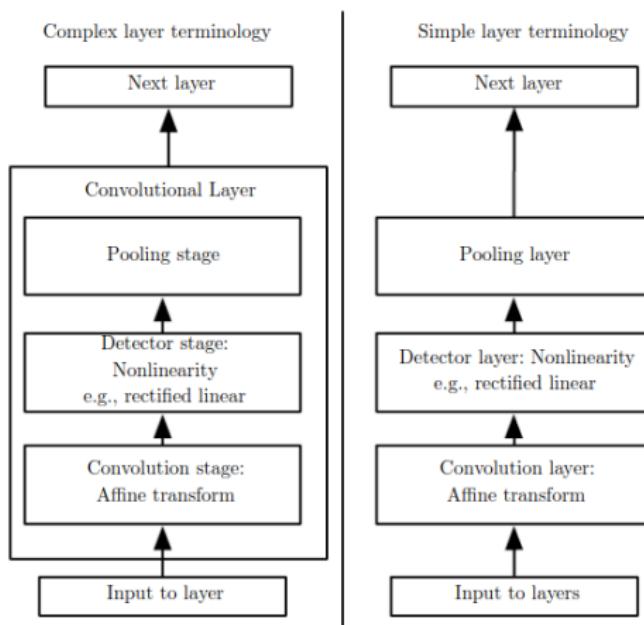


Figura: Terminología de capas complejas(izquierda) y de capas simples(derecha)

Fuente: (Goodfellow et al., 2016)

Arquitectura del Modelo

- La arquitectura de la red está inspirada en el arquitectura Inception (Szegedy et al., 2014) y en la arquitectura AlexNet(Krizhevsky et al., 2012) para la clasificación de imágenes. En la arquitectura Inception, el modelo creado es denominado GoogLeNet, similar a AlexNet donde varios módulos iniciales son apilados uno sobre el otro para producir el resultado final. En el módulo de inicio, en ese tipo de red se usaron diversos tamaños de filtros convolucionales para capturar características de diferente abstracción.

Arquitectura del Modelo

- La arquitectura de la red está inspirada en el arquitectura Inception (Szegedy et al., 2014) y en la arquitectura AlexNet(Krizhevsky et al., 2012) para la clasificación de imágenes. En la arquitectura Inception, el modelo creado es denominado GoogLeNet, similar a AlexNet donde varios módulos iniciales son apilados uno sobre el otro para producir el resultado final. En el módulo de inicio, en ese tipo de red se usaron diversos tamaños de filtros convolucionales para capturar características de diferente abstracción.
- El alto nivel de abstracción se captura con filtros de mayor tamaño y el de un nivel inferior con filtros de menor tamaño. Procesando información visual a diferentes escalas y al concatenarlas se obtiene un nivel eficiente de abstracción.

Arquitectura del Modelo

- En esta investigación se utilizó una versión modificada de las arquitecturas antes mencionadas. Se incorporó **funciones de escala-múltiple** (Sermanet and LeCun, 2011), lo que significa que la salida de las capas convolucionales no solo se envía a la capa posterior, sino que también se ramifica y se introduce al clasificador (capa totalmente conectada). La razón detrás de esto es que cuando el clasificador está tomando una decisión basada en convoluciones, podría encontrar que la salida de la primera o segunda capa convolucional también es útil. Básicamente con las características de escala múltiple depende del clasificador qué nivel de abstracción usar, ya que tiene acceso a las salidas de todas las capas convolucionales, es decir, características en todos los niveles de abstracción.

Arquitectura del Modelo

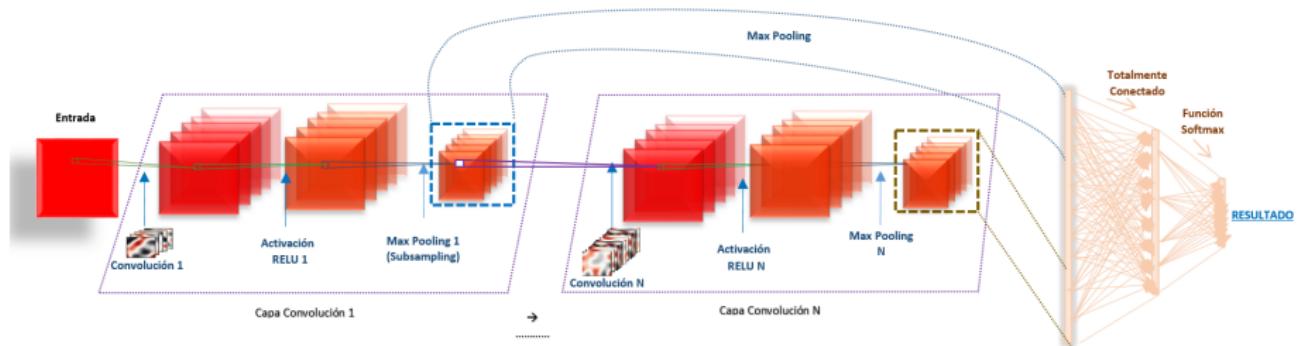


Figura: Modelo del diseño de la Red propuesta

Fuente propia

Componentes del Modelo

- Capa Convolucional

Componentes del Modelo

- Capa Convolucional
- Capa de Activación ReLU(Rectified Linear Units)

Componentes del Modelo

- Capa Convolucional
- Capa de Activación ReLU(Rectified Linear Units)
- Capa de Agrupación(Pooling)

Componentes del Modelo

- Capa Convolucional
- Capa de Activación ReLU(Rectified Linear Units)
- Capa de Agrupación(Pooling)
- Capa totalmente conectada (Fully-connected layer)

Capa Convolucional

- Los parámetros de la capa convolucional consisten básicamente en dos datos. La entrada y todo lo que respecta a un conjunto de filtros(también denominados kernels) cuyos valores se aprenden, es decir, empiezan con datos aleatorios y conforme avance el entrenamiento se van alterando.

Capa Convolucional

- Los parámetros de la capa convolucional consisten básicamente en dos datos. La entrada y todo lo que respecta a un conjunto de filtros(también denominados kernels) cuyos valores se aprenden, es decir, empiezan con datos aleatorios y conforme avance el entrenamiento se van alterando.
- Durante el proceso hacia adelante, se desliza (más precisamente, convolve) cada filtro a través del ancho y alto del volumen de entrada para calcular los productos de puntos entre las entradas del filtro y la entrada en cualquier posición.

Capa Convolucional

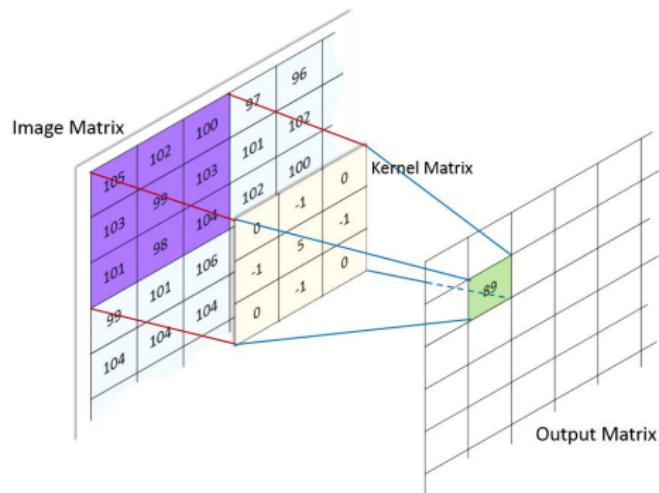


Figura: Posicionamiento del kernel/filtro por pixel

Fuente: Fuente Propia

Capa Convolucional

- A medida que deslizamos el filtro sobre el ancho y la altura del volumen de entrada produciremos un mapa de activación bidimensional que proporciona las respuestas de ese filtro en cada posición espacial. Por lo que simplemente se multiplica cada píxel en el filtro por el valor del píxel en la imagen. Para luego, sumar las respuestas y dividirlas por el número total de píxeles en el filtro.

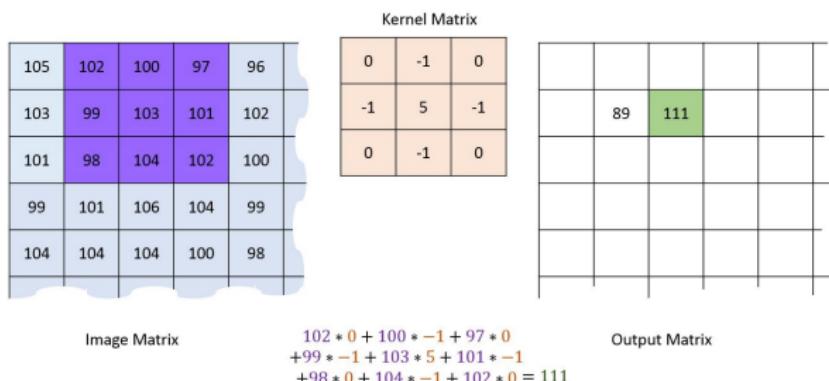


Figura: Cálculo Convolucional

Fuente: Fuente Propia

Capa Convolucional

Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:

Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:
 - ▶ Interacciones dispersas

Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:
 - ▶ Interacciones dispersas
 - ▶ Uso compartido de parámetros

Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:
 - ▶ Interacciones dispersas
 - ▶ Uso compartido de parámetros
 - ▶ Representaciones equivalentes

Capa Convolucional

- Capa Convolucional - Interacciones dispersas

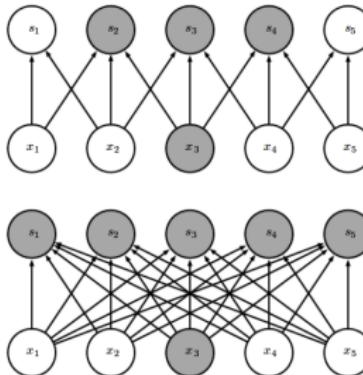


Figura: Conectividad dispersa vs No dispersa
Fuente:(Rohrer, 2016)

Cuando \mathbf{s} está formado por convolución con un kernel de ancho 3, solo tres salidas se ven afectadas por \mathbf{x} . (Abajo) Cuando \mathbf{s} está formado por la multiplicación de la matriz, la conectividad ya no es dispersa, por lo que todos los resultados se ven afectados por x_3 .

Capa Convolucional

- Capa Convolucional - Interacciones dispersas

Capa Convolucional

- **Capa Convolucional - Interacciones dispersas**

- ▶ Esto significa que necesitamos almacenar menos parámetros, lo que reduce los requisitos de memoria del modelo y mejora su eficiencia estadística. También significa que el cálculo de la salida requiere menos operaciones, (Goodfellow et al., 2016).

Capa Convolucional

- Capa Convolucional - Uso compartido de parámetros

Capa Convolucional

- **Capa Convolucional - Uso compartido de parámetros**

- ▶ En una red neuronal convolucional, cada miembro del kernel se utiliza en cada posición de la entrada (excepto tal vez algunos de los píxeles de los bordes, dependiendo de las decisiones de diseño con respecto al límite). El uso compartido de parámetros utilizado por la operación de convolución significa que en lugar de aprender conjuntos de parámetros para cada ubicación por separado, aprendemos un solo conjunto por kernel, (Goodfellow et al., 2016).

Capa Convolucional

- Capa Convolucional - Representaciones equivalentes

Capa Convolucional

- **Capa Convolucional - Representaciones equivalentes**

- ▶ La forma particular de compartir los parámetros hace que la capa tenga una propiedad llamada **representaciones equivalentes**. Decir que una función es equivalente significa que si la entrada cambia, la salida cambia de la misma manera.

Capa Convolucional

- **Capa Convolucional - Representaciones equivalentes**

- ▶ La forma particular de compartir los parámetros hace que la capa tenga una propiedad llamada **representaciones equivalentes**. Decir que una función es equivalente significa que si la entrada cambia, la salida cambia de la misma manera.
- ▶ La convolución crea un mapa en 2-D de donde aparecen ciertas características en la entrada. Si movemos el objeto en la entrada, su representación se moverá la misma cantidad en la salida.

Capa Convolucional

- Capa Convolucional - Representaciones equivalentes

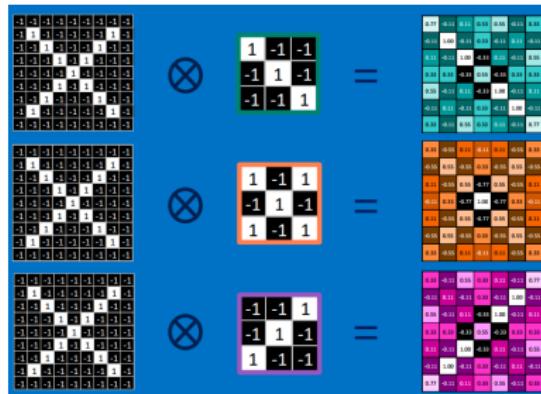


Figura: Resultado de Convolución(conjunto de mapas de activación. generado a partir de 3 filtros para 3 características: diagonal derecha, cruzamiento central y diagonal izquierda). Simbolo de convolución: \otimes

Fuente: (Rohrer, 2016)

Capa Convolucional

• Capa Convolucional - Construcción de un filtro o kernel

- 1 La extensión espacial es el tamaño del filtro, comúnmente es de tamaño impar tanto en largo y ancho.
- 2 El stride es otra pieza del bloque de construcción básico de los filtros convolucionales. Este representa el 'paso' en la operación de convolución indicando cuánto es que se debe desplazar un filtro en una imagen con cada paso. El filtro se desliza sobre la imagen, se detiene en cada longitud de salto y realiza las operaciones necesarias en ese paso.

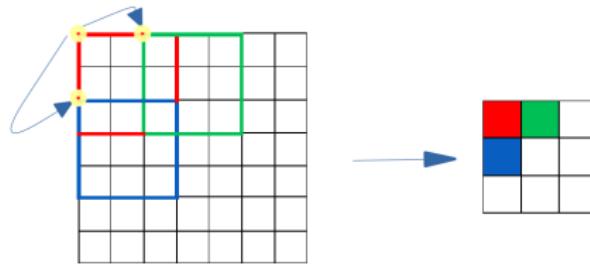


Figura: Imagen con stride igual a 2, para el filtro tanto en largura como anchura

Fuente propia

Capa Convolucional

• Capa Convolucional - Construcción de un filtro o kernel

- 3 Zero-padding agrega ceros alrededor del borde de una imagen.

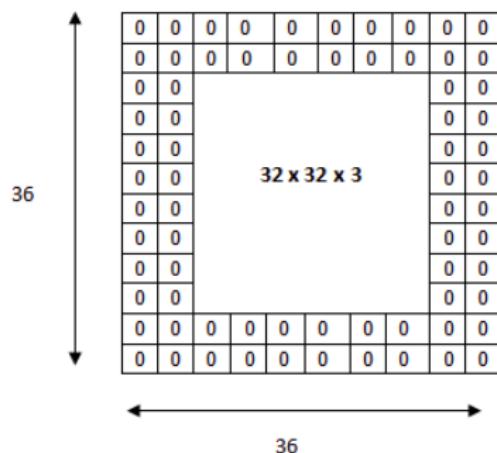


Figura: Ejemplo de zero-padding con tamaño 2

Fuente propia

Capa Convolucional

- Capa Convolucional - Ecuación de Convolución

Capa Convolucional

- **Capa Convolucional - Ecuación de Convolución**

$$\bullet \text{ conv}_j^n = \sum_{k=1}^k x_k^n \times w_{kj}^n + b_n$$

En el que:

- ▶ x son valores de entrada
- ▶ w, b pesos y biases(sesgos) del kernel, respectivamente
- ▶ n es el número de la capa
- ▶ j es el número del filtro de salida
- ▶ k es la cantidad de filtros en la capa $n - 1$ o n

Capa Convolucional

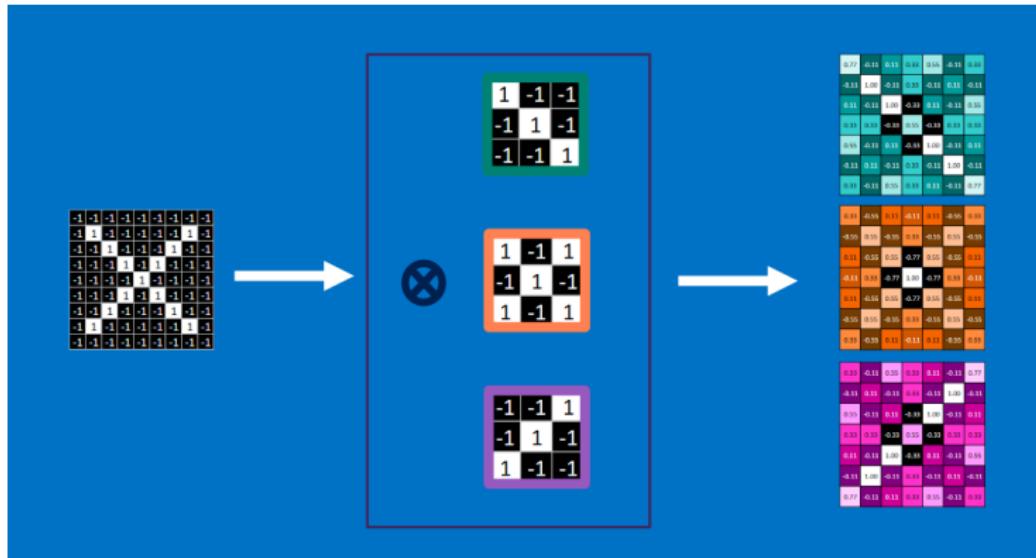


Figura: Resultado Final de Convolución

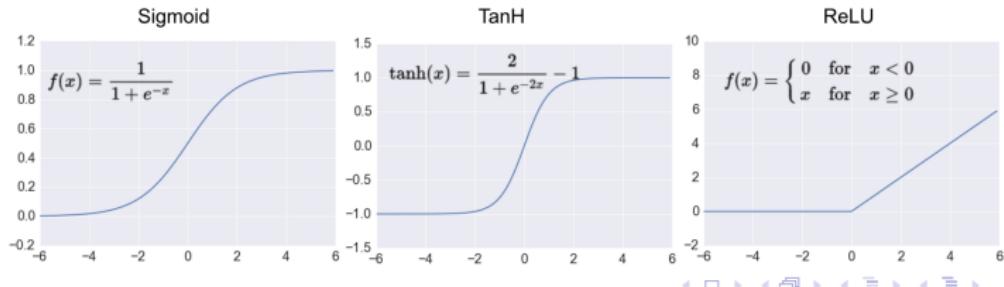
Fuente: (Rohrer, 2016)

Capa ReLU(Rectified Linear Units)

- Una red neuronal artificial consiste básicamente en multiplicaciones y suma de matrices. Si solo utilizáramos estos cálculos lineales, podríamos apilarlos uno encima del otro y esa no sería una red muy profunda. Por lo tanto, a menudo se utiliza funciones de activación no lineales en cada capa de la red.

Capa ReLU(Rectified Linear Units)

- Una red neuronal artificial consiste básicamente en multiplicaciones y suma de matrices. Si solo utilizáramos estos cálculos lineales, podríamos apilarlos uno encima del otro y esa no sería una red muy profunda. Por lo tanto, a menudo se utiliza funciones de activación no lineales en cada capa de la red.
- Estas son las tres funciones de activación no lineal más populares:
 - 1) Sigmoid (analiza un valor entre 0 y 1)
 - 2) TanH (analiza un valor entre -1 y 1)
 - 3) ReLU (si el valor es negativo, se convierte en 0, de lo contrario, permanece igual)



Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.

Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.
 - La red puede entrenar mucho más rápido (debido a la eficiencia computacional) sin hacer una diferencia significativa en la precisión.

Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.
 - La red puede entrenar mucho más rápido (debido a la eficiencia computacional) sin hacer una diferencia significativa en la precisión.
 - Alivia el problema del gradiente de fuga, que es el problema donde las capas inferiores de la red entranan muy lentamente porque el gradiente de optimización disminuye exponencialmente a través de las capas.

Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.
 - ▶ La red puede entrenar mucho más rápido (debido a la eficiencia computacional) sin hacer una diferencia significativa en la precisión.
 - ▶ Alivia el problema del gradiente de fuga, que es el problema donde las capas inferiores de la red entran muy lentamente porque el gradiente de optimización disminuye exponencialmente a través de las capas.
- El hecho de que entradas en la función de activación de valores menores o iguales a cero resulten cero, induce a la dispersión en las unidades ocultas, que según lo comentado anteriormente, produce representaciones dispersas las cuales se consideran más valiosas, (Nair and Hinton, 2010).

Capa ReLU(Rectified Linear Units)

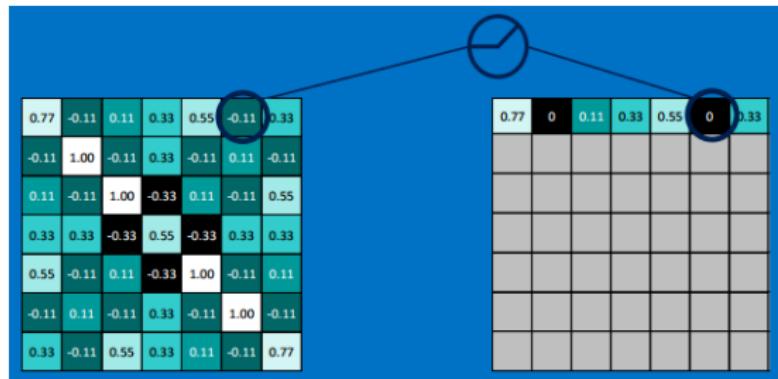


Figura: Procedimiento de la función ReLU

Fuente: (Rohrer, 2016)

Capa ReLU(Rectified Linear Units)

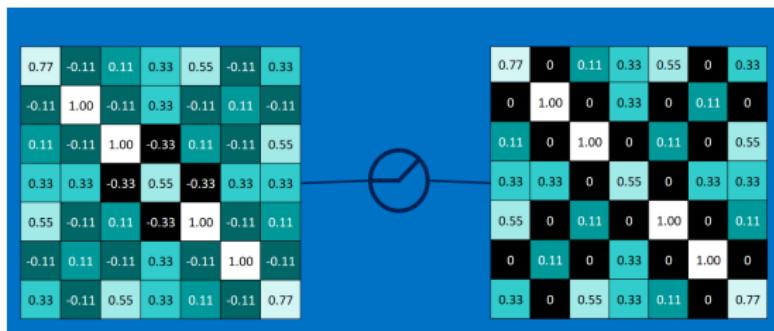


Figura: Procedimiento de la función ReLU

Fuente: (Rohrer, 2016)

Capa ReLU(Rectified Linear Units)



Figura: Procedimiento de la función ReLU
Fuente: (Rohrer, 2016)

Técnica Dropout

- Es una técnica de regularización que tiene por objetivo de reducir el sobreajuste que puede darse durante el entrenamiento de una red neuronal. Consiste en establecer a cero la salida de cada neurona oculta con una probabilidad 0.5(comúnmente). Las neuronas que se abandonan de esta manera no contribuyen al pase directo y no participan en las siguientes etapas de entrenamiento.

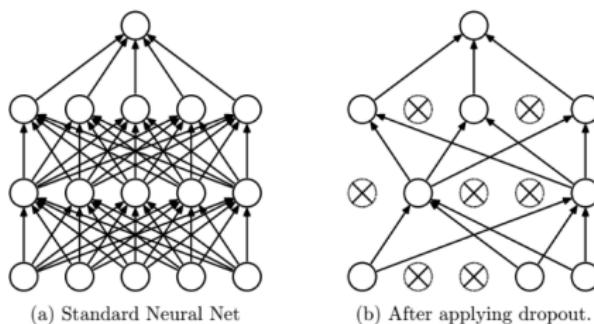


Figura: En la izquierda la red neuronal común y a la derecha la red neuronal diluida producida por la aplicación de dropout

(Romero, 2015)

Técnica Dropout

- Por lo tanto, cada vez que se presenta una entrada, la red neuronal muestra una arquitectura diferente, pero todas estas arquitecturas comparten ponderaciones. Esta técnica reduce las coadaptaciones complejas de las neuronas, ya que una neurona no puede depender de la presencia de otras neuronas particulares. Por lo tanto, se ve forzado a aprender características más robustas que son útiles en conjunción con muchos subconjuntos aleatorios diferentes de las otras neuronas.

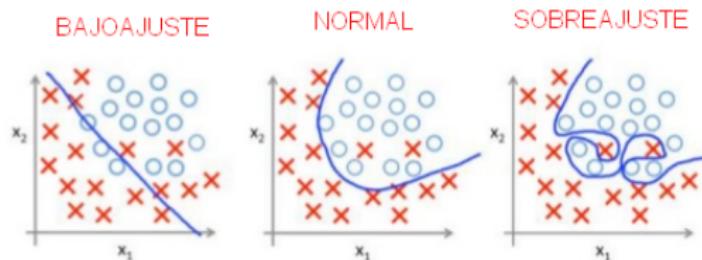


Figura: Procesos que pueden ocurrir durante entrenamiento. Dropout evita el sobreajuste

Fuente propia

Capa de Agrupación(Pooling)

- Una variedad de cálculos que reducen la dimensionalidad de un mapa de características se conocen como agrupación. La agrupación, que se aplica a cada canal por separado, permitiendo que la red sea robusta e invariante a pequeños cambios y distorsiones.

Capa de Agrupación(Pooling)

- Una variedad de cálculos que reducen la dimensionalidad de un mapa de características se conocen como agrupación. La agrupación, que se aplica a cada canal por separado, permitiendo que la red sea robusta e invariante a pequeños cambios y distorsiones.
- La capa Pooling(también conocida como una capa de reducción de resolución) combina o agrupa, un conjunto de valores en su campo receptivo en un menor número de valores.

Capa de Agrupación(Pooling)

- Una variedad de cálculos que reducen la dimensionalidad de un mapa de características se conocen como agrupación. La agrupación, que se aplica a cada canal por separado, permitiendo que la red sea robusta e invariante a pequeños cambios y distorsiones.
- La capa Pooling(también conocida como una capa de reducción de resolución) combina o agrupa, un conjunto de valores en su campo receptivo en un menor número de valores.
- Puede ser configurado en función del tamaño de su campo receptivo (por ejemplo, 2×2) y en función a la operación de agrupamiento (por ejemplo, máximo-max o promedio-average).

| 2x2 pooling, stride 2 | | | |
|-----------------------|----|----|---|
| 9 | 3 | 5 | 3 |
| 10 | 32 | 2 | 2 |
| 1 | 3 | 21 | 9 |
| 2 | 6 | 11 | 7 |

Max pooling **Average pooling**

| 32 | 5 |
|----|----|
| 6 | 21 |

| 18 | 3 |
|----|----|
| 3 | 12 |

Figura: Formas de agrupamiento(pooling)

Fuente: (Jia et al., 2014)

Capa de Agrupación(Pooling)

- El Max-pooling es la más popular reduciendo progresivamente el tamaño espacial de la representación de una imagen(mapa de características) mientras conserva la información más importante en ella. El proceso matematico consiste en pasar una pequeña ventana(kernel)através de una imagen y tomar el valor máximo de la ventana en cada paso.

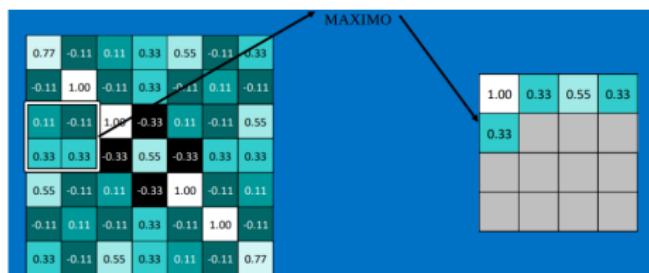


Figura: Operación MAX-pooling en capa de Agrupación

Fuente: (Rohrer, 2016)

Capa de Agrupación(Pooling)

- Este proceso es aplicado para cada mapa de activación(salida de la capa de convolución). Análogamente con la capa de convolución, el resultado en esta capa es un conjunto de imágenes que muestran una versión agrupada de las imágenes de entrada.



Figura: Resultado de Agrupación

Fuente: (Rohrer, 2016)

Componentes del Modelo - Capas Conectadas

- Finalmente la salida de cada capa se convierte en entrada para una posterior.

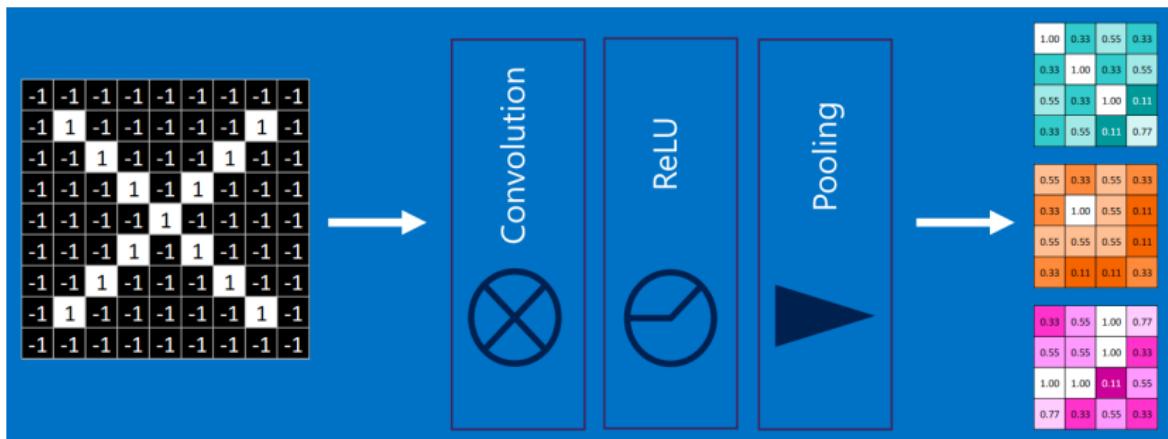


Figura: Capas Apiladas

Fuente: (Rohrer, 2016)

Componentes del Modelo - Capas Conectadas

- Las capas pueden ser repetidas múltiples veces.

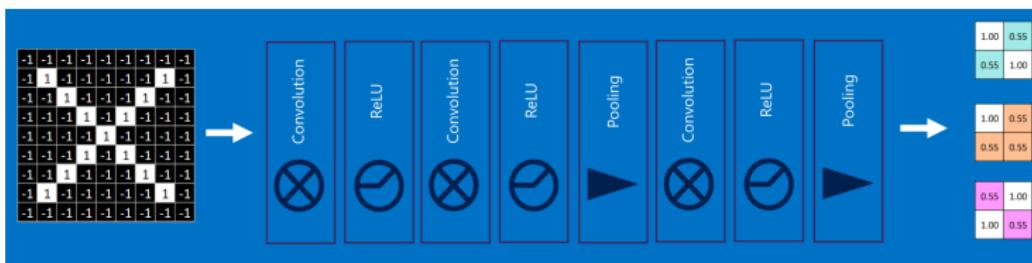


Figura: Apilación Profunda de Capas

Fuente: (Rohrer, 2016)

Capa totalmente conectada (Fully-connected layer)

- Esta capa por lo general aparece al final de la arquitectura y es similar al Perceptrón multicapa (Multilayer Perceptron -MLP), en el cual la neurona de salida se conecta a todas las neuronas de entrada y el peso de las conexiones son actualizadas usando el método de retropropagación.

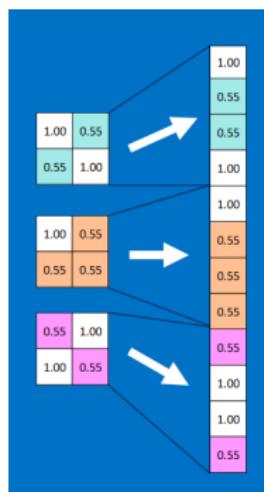


Figura: Eventualmente, con un mapa de características lo suficientemente pequeño, el contenido se aplastará en un vector de una sola dimensión y será entrada para en un MLP totalmente conectado para su procesamiento

Fuente: (Rohrer, 2016)

Capa totalmente conectada (Fully-connected layer)

- En esta última capa del modelo generalmente se utiliza una función de activación diferente, porque en esta capa se pretende que tenga una salida determinada. La **función softmax**, conocida también como función exponencial normalizada, es muy popular cuando se hace la clasificación.

Capa totalmente conectada (Fully-connected layer)

- En esta última capa del modelo generalmente se utiliza una función de activación diferente, porque en esta capa se pretende que tenga una salida determinada. La **función softmax**, conocida también como función exponencial normalizada, es muy popular cuando se hace la clasificación.
- La salida de una red neuronal completamente conectada no es una distribución de probabilidad, sin embargo, el uso de esta función ayuda a obternerla. Por lo que esta función permitirá estimar la probabilidad de que la imagen de entrada pertenezca a cada una de las clases al realizar una normalización con el objetivo que el valor de cada neurona esté limitado entre cero y uno y así permitir que el resultado de las neuronas de salida sumen a uno, (Bishop, 2006).

Validación Cruzada

- La cantidad de errores en la votación, nos indica qué tan buenas son nuestras pesos y bias. Las características y los pesos se pueden ajustar para reducir el error. Cada valor se ajusta un poco más alto y un poco más bajo, y el nuevo error se calcula cada vez.

Validación Cruzada

- La cantidad de errores en la votación, nos indica qué tan buenas son nuestras pesos y bias. Las características y los pesos se pueden ajustar para reducir el error. Cada valor se ajusta un poco más alto y un poco más bajo, y el nuevo error se calcula cada vez.
- Después de hacer esto para cada píxel de función en cada capa convolucional y cada peso en cada capa totalmente conectada, los nuevos pesos dan una respuesta que funciona ligeramente mejor para esa imagen, (Bishop, 2006).

Validación Cruzada

- La cantidad de errores en la votación, nos indica qué tan buenas son nuestras pesos y bias. Las características y los pesos se pueden ajustar para reducir el error. Cada valor se ajusta un poco más alto y un poco más bajo, y el nuevo error se calcula cada vez.
- Después de hacer esto para cada píxel de función en cada capa convolucional y cada peso en cada capa totalmente conectada, los nuevos pesos dan una respuesta que funciona ligeramente mejor para esa imagen, (Bishop, 2006).
- Esto se repite con cada imagen subsiguiente en el conjunto de imágenes etiquetadas. Los problemas que ocurren en una sola imagen se olvidan rápidamente, pero los patrones que aparecen en muchas imágenes se integran en las características y los pesos de conexión.

Validación Cruzada

- Para el entrenamiento, el conjunto de datos se divide en un subconjunto de entrenamiento y otro para validación. En esta investigación se dividirá el 75 % para entrenamiento y 75 % para validación.

Validación Cruzada

- Para el entrenamiento, el conjunto de datos se divide en un subconjunto de entrenamiento y otro para validación. En esta investigación se dividirá el 75 % para entrenamiento y 75 % para validación.
- La división en subgrupos de entrenamiento y validación generalmente se realiza de manera aleatoria, para garantizar que ambos subconjuntos sean muestras aleatorias de la misma distribución.

Validación Cruzada

- Para una red neuronal, normalmente verá la ecuación escrita en una forma donde y es el vector de verdad y la variable \hat{y} (o algún otro valor tomado directamente de la última salida de la capa) es la estimación, y se vería así para un solo ejemplo:

$$L = -\mathbf{y} \cdot \log(\hat{\mathbf{y}})$$

Validación Cruzada

- Para una red neuronal, normalmente verá la ecuación escrita en una forma donde y es el vector de verdad y la variable \hat{y} (o algún otro valor tomado directamente de la última salida de la capa) es la estimación, y se vería así para un solo ejemplo:

$$L = -\mathbf{y} \cdot \log(\hat{\mathbf{y}})$$

- A menudo esta ecuación es promediada sobre todos los ejemplos como una función de costo. No siempre se cumple estrictamente en las descripciones, pero generalmente una función de pérdida es de nivel inferior y describe cómo una sola instancia o componente determina un valor de error, mientras que una función de costo es de nivel superior y describe cómo se evalúa un sistema completo para la optimización.

Validación Cruzada

- Para una red neuronal, normalmente verá la ecuación escrita en una forma donde y es el vector de verdad y la variable \hat{y} (o algún otro valor tomado directamente de la última salida de la capa) es la estimación, y se vería así para un solo ejemplo:

$$L = -\mathbf{y} \cdot \log(\hat{\mathbf{y}})$$

- A menudo esta ecuación es promediada sobre todos los ejemplos como una función de costo. No siempre se cumple estrictamente en las descripciones, pero generalmente una función de pérdida es de nivel inferior y describe cómo una sola instancia o componente determina un valor de error, mientras que una función de costo es de nivel superior y describe cómo se evalúa un sistema completo para la optimización.
- Una función de costo basada en pérdida de clasificación multiclas para un conjunto de datos de tamaño N podría verse así:

$$J = -\frac{1}{N} (\sum_{i=1}^N \mathbf{y}_i \cdot \log(\hat{\mathbf{y}}_i))$$

Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.

Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.
- En el caso del entrenamiento de una red neuronal, el gradiente simplemente mide el cambio en todos los pesos con respecto al cambio en el error.

Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.
- En el caso del entrenamiento de una red neuronal, el gradiente simplemente mide el cambio en todos los pesos con respecto al cambio en el error.
- El descenso de gradiente es un algoritmo de optimización iterativa utilizado al entrenar un modelo de aprendizaje automático, basado en una función convexa, que ajusta sus parámetros iterativamente para minimizar la función de pérdida(error) hasta su mínimo local.(Dong and Zhou, 2008)

Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.
- En el caso del entrenamiento de una red neuronal, el gradiente simplemente mide el cambio en todos los pesos con respecto al cambio en el error.
- El descenso de gradiente es un algoritmo de optimización iterativa utilizado al entrenar un modelo de aprendizaje automático, basado en una función convexa, que ajusta sus parámetros iterativamente para minimizar la función de pérdida(error) hasta su mínimo local.(Dong and Zhou, 2008)
- La idea detrás del descenso de gradiente es disminuir de forma gradual, pero constante, el error de salida ajustando los pesos. Intuitivamente, se conoce que si un cambio en un peso aumentará (disminuirá) el error, entonces queremos disminuir (aumentar) ese peso. Matemáticamente, representa el cambio en el error dado un cambio de unidad en el peso:

$$\frac{\partial E}{\partial w_{ij}}$$

La derivada del error con respecto al peso

Descenso de Gradiente

- Una vez que encontremos esta derivada, actualizamos el peso a través de lo siguiente:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Representación de la distancia por la dirección del cambio. η es tasa de aprendizaje

Descenso de Gradiente

- Una vez que encontramos esta derivada, actualizamos el peso a través de lo siguiente:

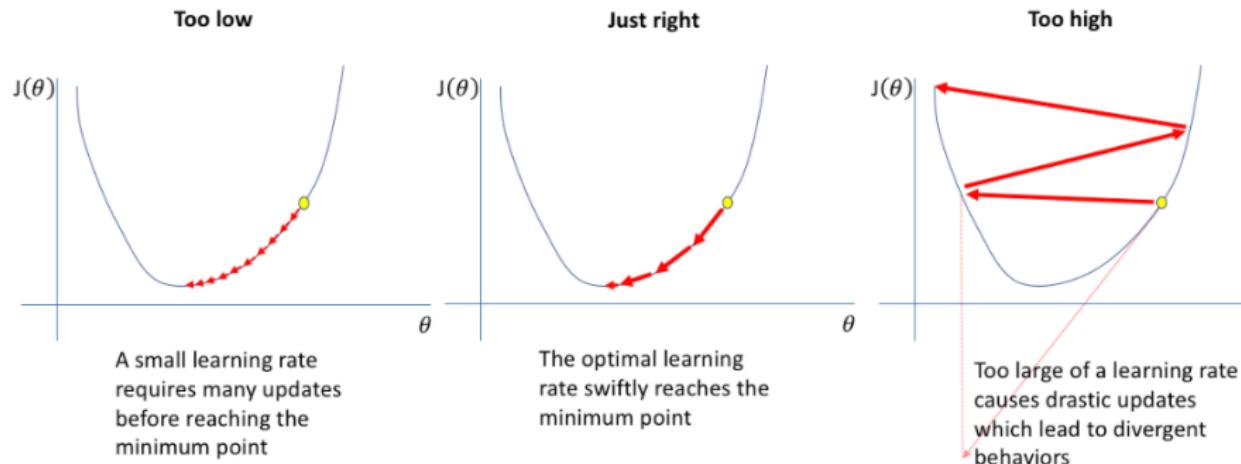
$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Representación de la distancia por la dirección del cambio. η es tasa de aprendizaje

- La tasa de Aprendizaje suele disminuir gradualmente durante las épocas de la fase de entrenamiento. Si actualizamos todos los pesos usando esta misma fórmula, esto equivale a moverse en la dirección de descenso más pronunciado a lo largo de la superficie de error, de ahí el nombre, descenso de gradiente.

Tasa de Aprendizaje (Learning Rate)

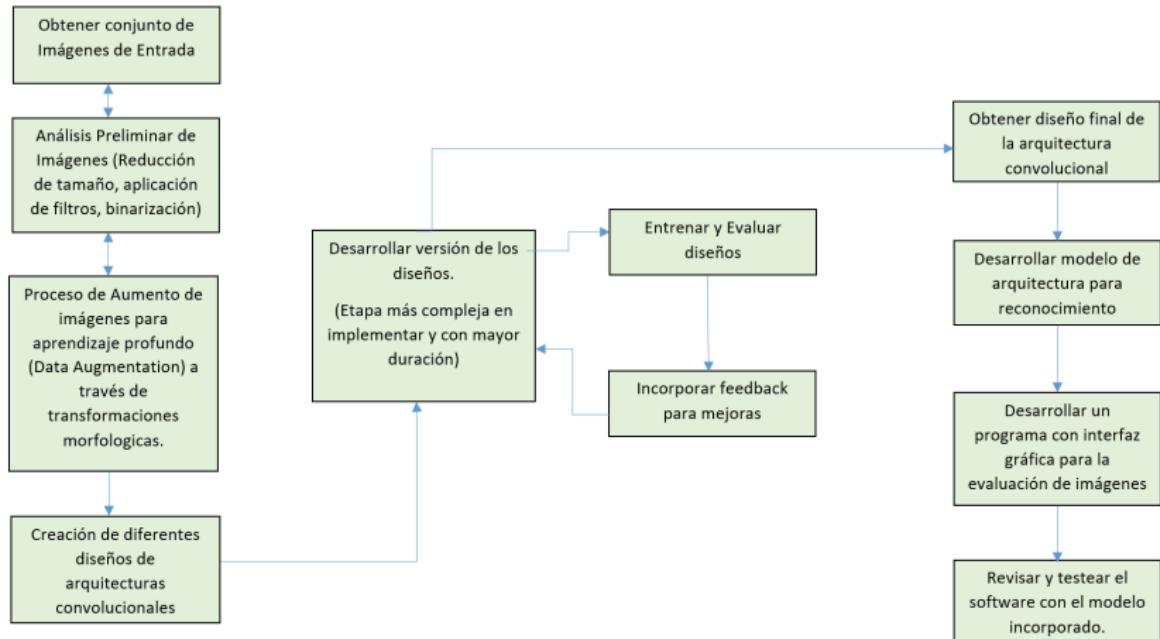
- Este parámetro puede entenderse como el tamaño del paso en la optimización para minimizar la función de pérdida de la red, es decir, es un parámetro que determina cuánto influye un paso de actualización en el valor actual de los pesos, por ejemplo, el tamaño del paso en el que el gradiente cae en la dirección máxima de la pendiente, (Jordan, 2018).



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

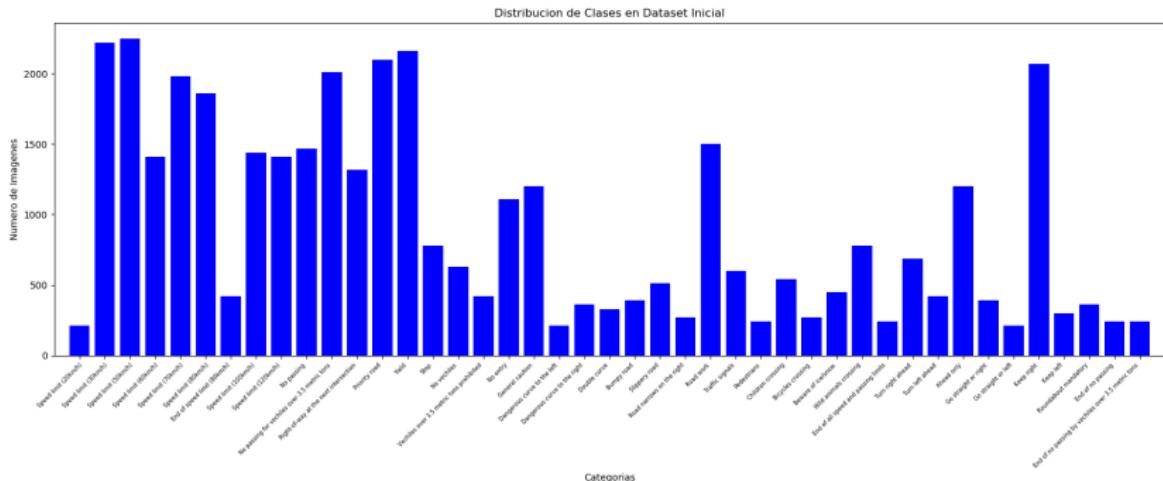
- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Diseño del Modelo del Ciclo de Vida del Desarrollo



Datos para el entrenamiento

• Señales de Tránsito de Alemania

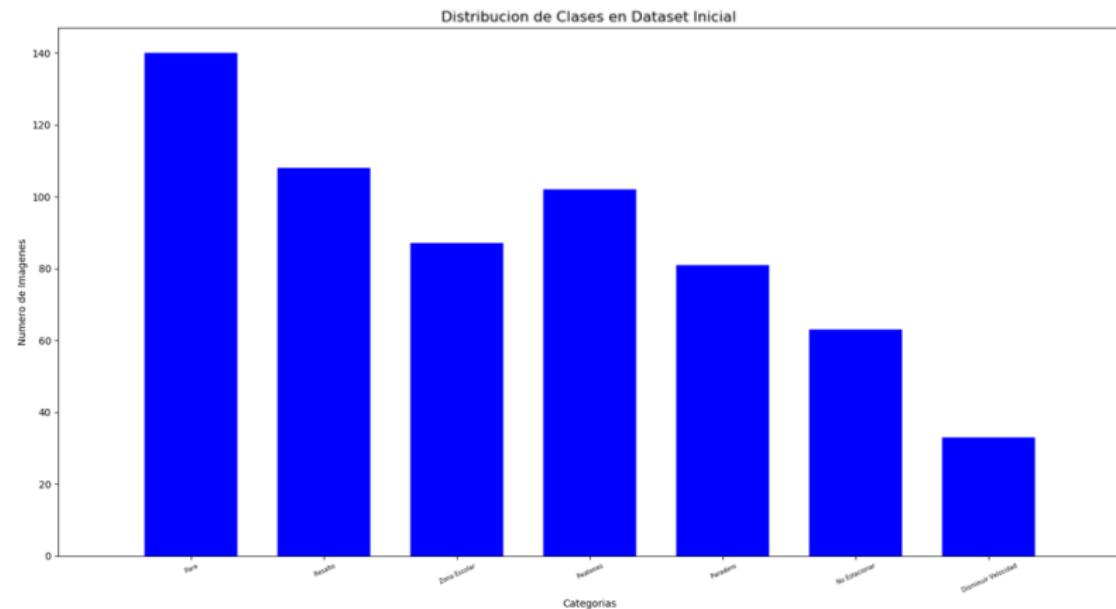


Distribución de ejemplos por señal para el entrenamiento (Total 39209)

Fuente propia

Datos para el entrenamiento

- Señales de Tránsito de Perú

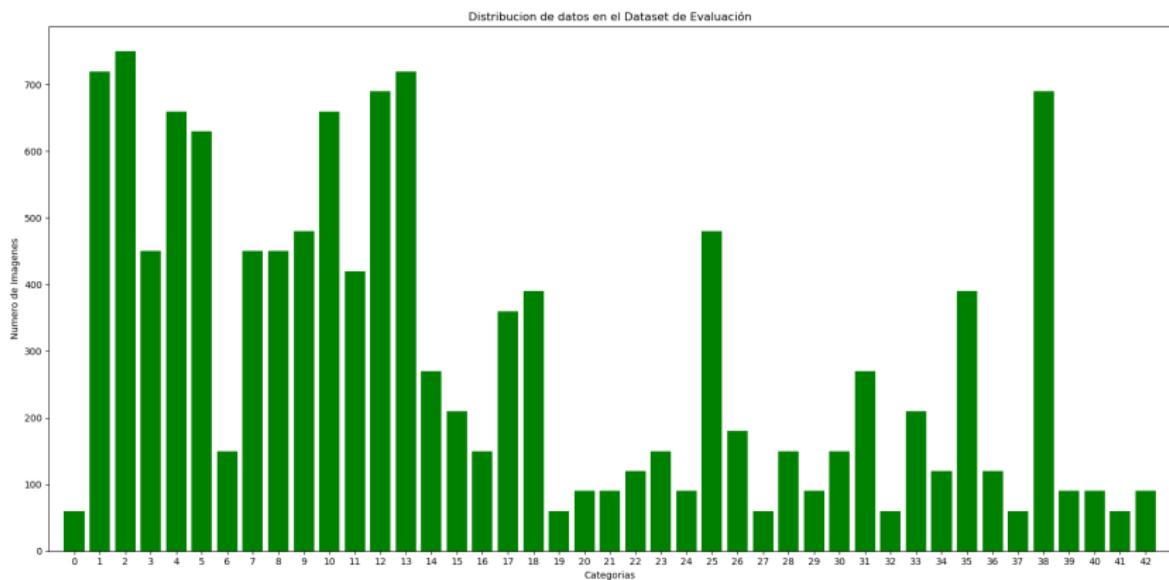


Distribución de ejemplos por señal para el entrenamiento (Total 614)

Fuente propia

Datos para la Evaluación

- Señales de Tránsito de Alemania (12630 imágenes)

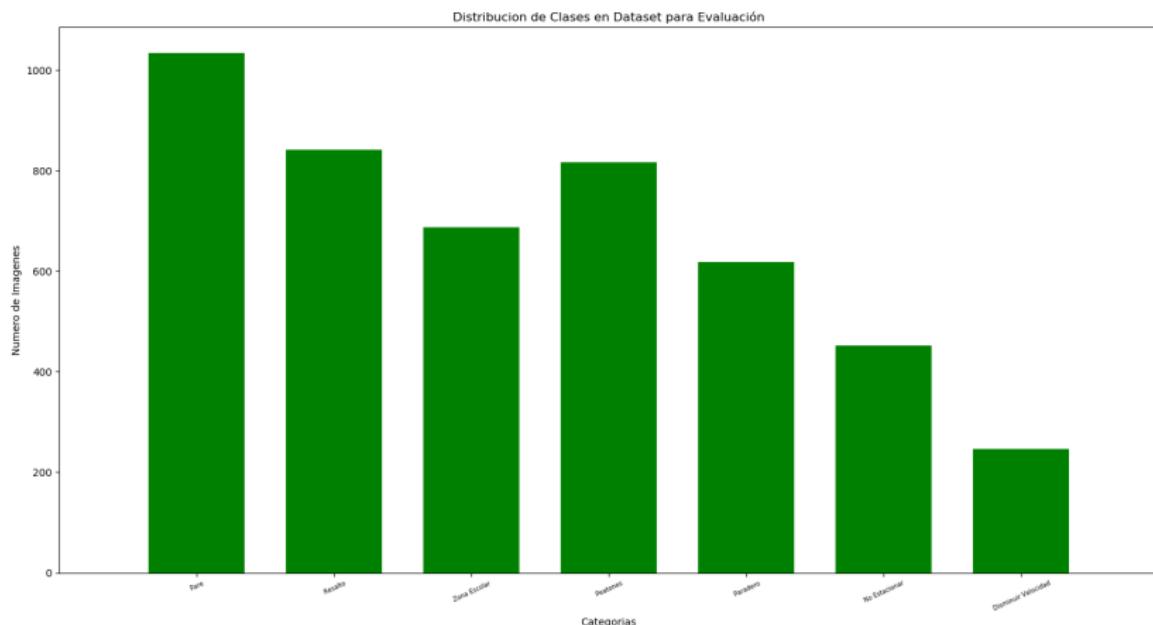


Distribución de ejemplos por señal para la evaluación - Alemania

Fuente propia

Datos para la Evaluación

- Señales de Tránsito de Perú (4698 imágenes)



Distribución de ejemplos por señal para la evaluación - Perú

Fuente propia

Proceso de Aumento de Datos(Data Augmentation)

1 Flipping

Flipping horizontal:

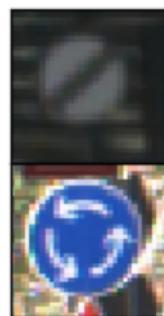
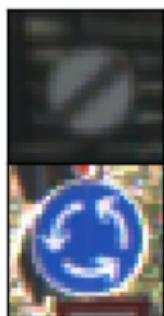


Flipping Vertical:



Proceso de Aumento de Datos(Data Augmentation)

Flipping Horizontal y Vertical:



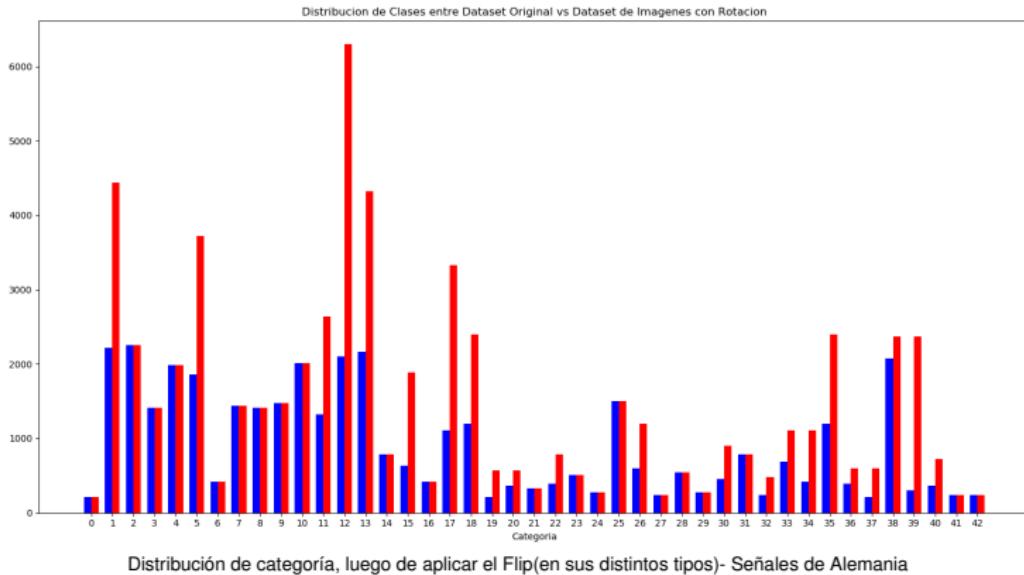
Proceso de Aumento de Datos(Data Augmentation)

Incluso, hay signos que luego de voltearse, deben clasificarse como un signo de alguna otra clase. Esto sigue siendo útil, ya que podemos utilizar los datos de estas clases para ampliar sus contrapartes.



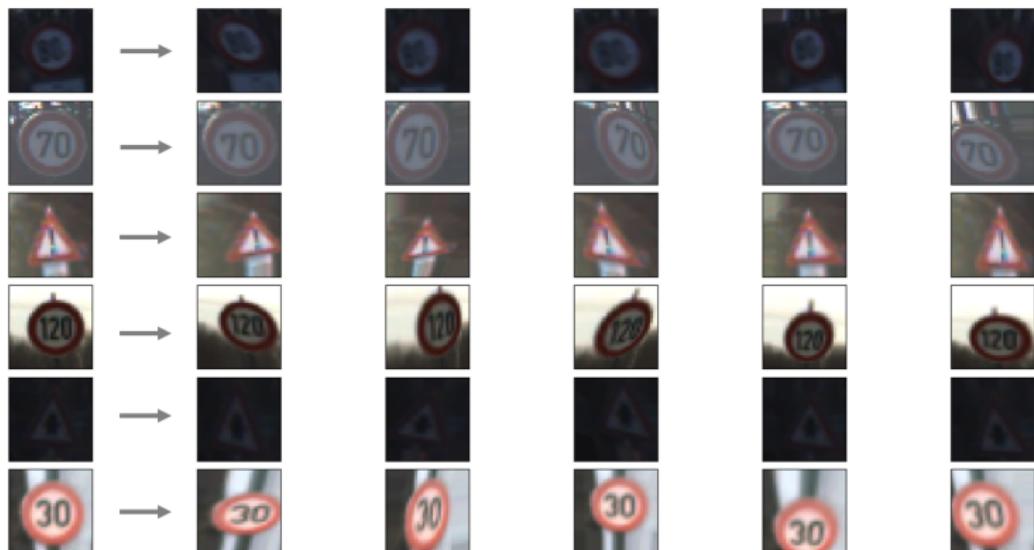
Proceso de Aumento de Datos(Data Augmentation)

Finalmente obtenemos una nueva distribución de datos luego de haber aplicado flipping a ciertas imágenes. Esta distribución consta de 63538 imágenes.



Proceso de Aumento de Datos(Data Augmentation)

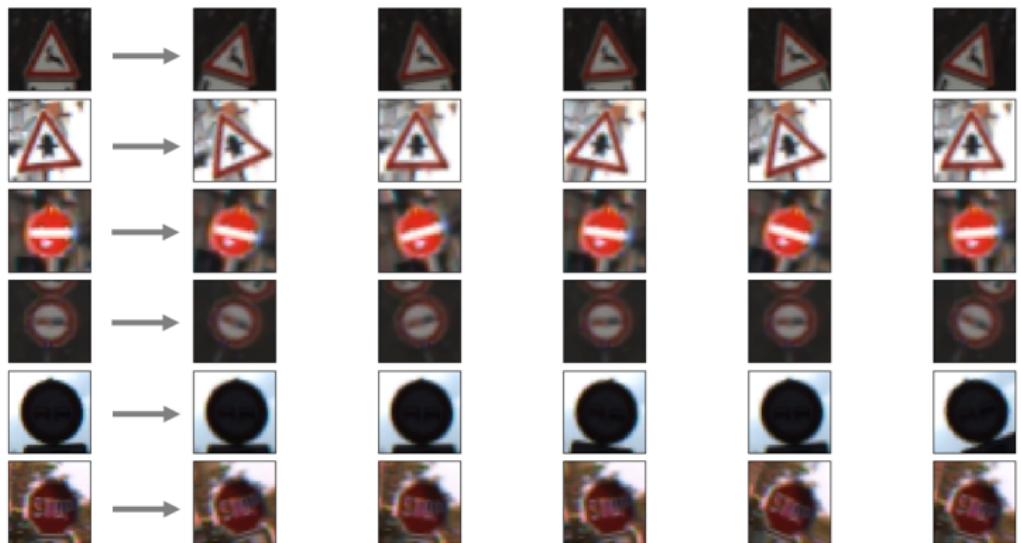
② Projection(Proyección)



Ejemplo de cinco proyecciones por cada imagen - Dataset Alemania

Proceso de Aumento de Datos(Data Augmentation)

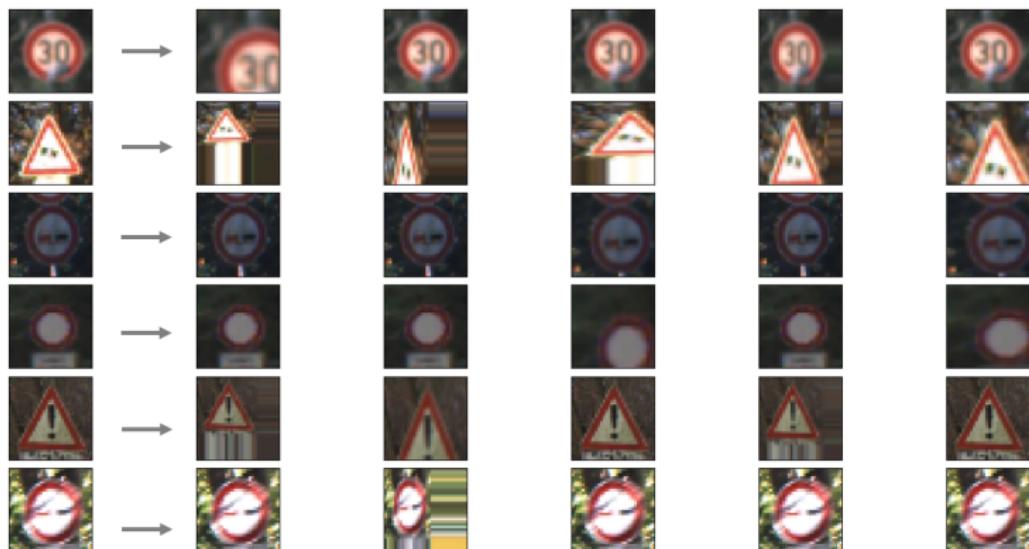
③ Rotation(Rotación)



Ejemplo de cinco rotaciones por cada imagen - Dataset Alemania

Proceso de Aumento de Datos(Data Augmentation)

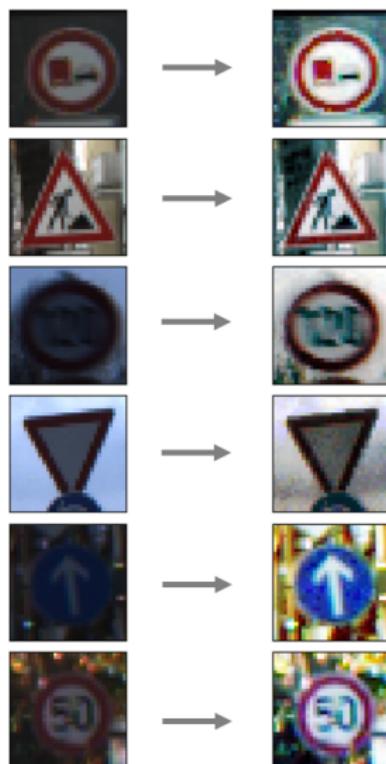
4 Zoom



Ejemplo de cinco aplicaciones de zoom(in/out) por cada imagen - Dataset Alemania

Proceso de Aumento de Datos(Data Augmentation)

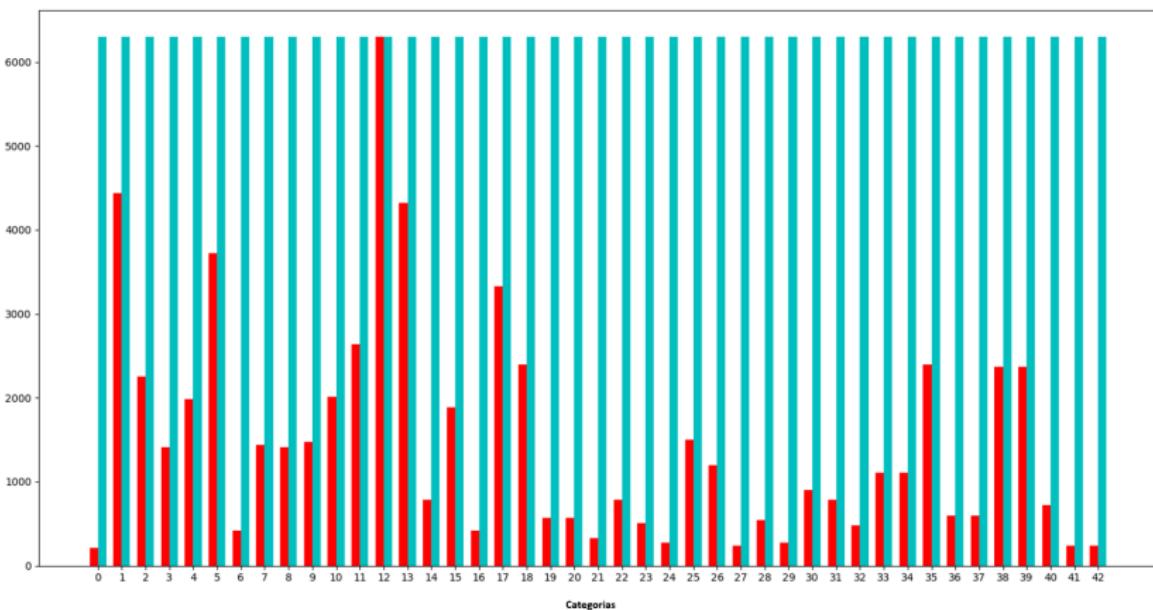
5 Equalizacion del histograma



Dataset final

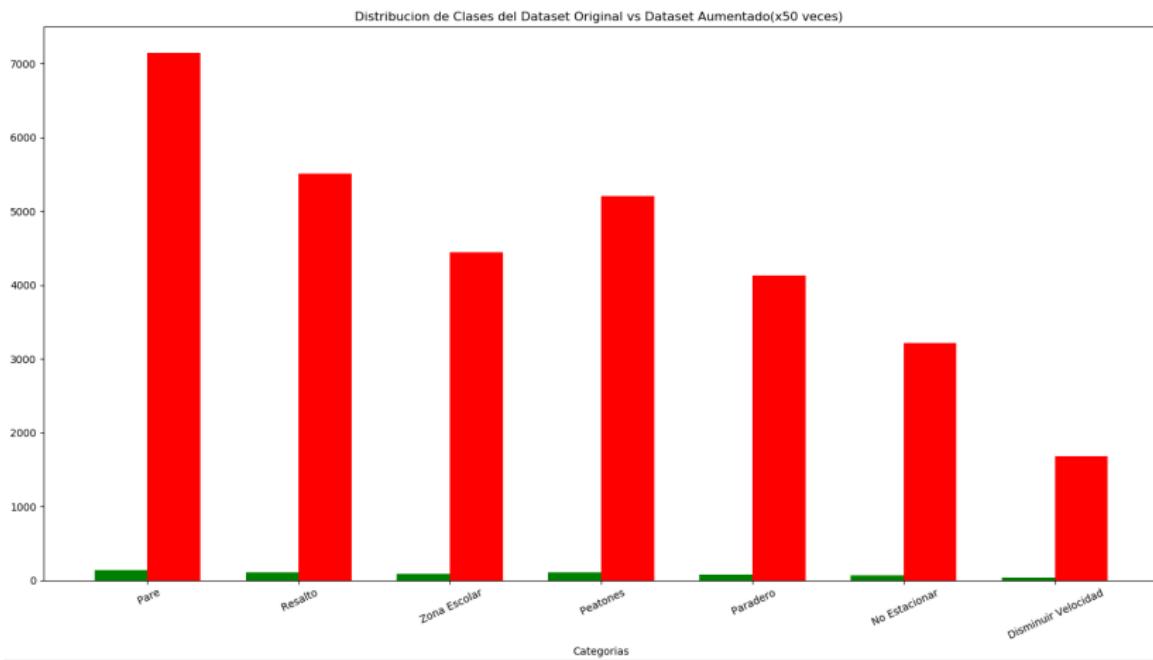
- Señales de Tránsito de Alemania (270900 imágenes)

Distribución de Clases Imágenes Rotadas vs Imágenes Aumentadas manteniendo un Balance



Dataset final

- Señales de Tránsito de Perú (31314 imágenes)



Pre-procesamiento de Imágenes(Normalization)

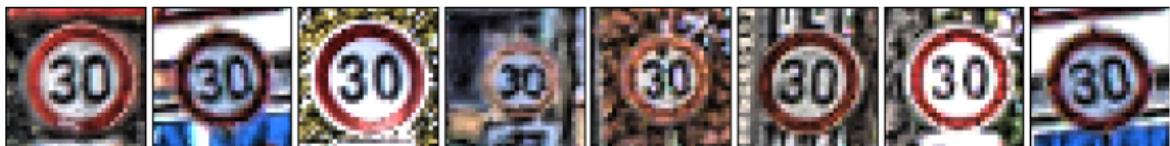
- La técnica utilizada CLAHE es un algoritmo para la mejora del contraste local, que utiliza histogramas calculados sobre diferentes regiones en una imagen.

Pre-procesamiento de Imágenes(Normalization)

- La técnica utilizada CLAHE es un algoritmo para la mejora del contraste local, que utiliza histogramas calculados sobre diferentes regiones en una imagen.
- Difiere de la ecualización de histograma ordinaria en el sentido de que el método adaptativo computa varios histogramas, cada uno correspondiente a una sección distinta de la imagen, y los utiliza para redistribuir los valores de luminosidad de la imagen.

Pre-procesamiento de Imágenes(Normalization)

- La técnica utilizada CLAHE es un algoritmo para la mejora del contraste local, que utiliza histogramas calculados sobre diferentes regiones en una imagen.
- Difiere de la ecualización de histograma ordinaria en el sentido de que el método adaptativo computa varios histogramas, cada uno correspondiente a una sección distinta de la imagen, y los utiliza para redistribuir los valores de luminosidad de la imagen.
- Mejora el nivel de visibilidad de una imagen o video con niebla ya que permite mejorar los detalles locales incluso en regiones que son más oscuras o más claras que la mayoría de regiones de la imagen. (Yadav et al., 2014)



Pre-procesamiento de Imágenes(Normalization)

- Pierre Sermanet y Yann LeCun mencionaron en su artículo (LeCun et al., 1998), que el uso de canales de color no pareció mejorar mucho las cosas. Además, debido a diversas condiciones o problemas de iluminación, no es adecuado procesar directamente las imágenes que se capturan a través de la cámara o sensores de imágenes, es por ello que en esta investigación **se usará un solo canal** en el modelo, es decir las imágenes estarán en escala de grises en lugar de tener 3 canales de colores.



Arquitectura del Modelo

- Los hiperparámetros engloban funciones, variables y constantes utilizadas durante la construcción de las diferentes arquitecturas; estas varían, sin embargo siguiendo conceptos teóricos, antecedentes (investigaciones previas) y sobre todo después de algunas pruebas realizadas, los siguientes hiperparámetros fueron seleccionados de manera específica:

| HIPERPARÁMETROS | TIPO | HIPERPARÁMETROS | TIPO |
|-----------------------------------|----------------------------------|---|------------------|
| Inicialización de Pesos | Xavier | Tasa de Aprendizaje | 0.0005 |
| Alg. de Optimización | Optimizador Adam | Método de Validación | Entropía Cruzada |
| Fun. Activ. Capas Convolucionales | RELU y DropOut | Fun. Activ. Capas Totalmente Conectadas | Func. Softmax |
| Método de Regularización | $L2$ Lasso($\lambda = 0.0001$) | Épocas | 100 |

Modelos Propuestos

1 MODELO A

Alemania

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|------------------------|---------------------|-----------------------------|----------|------------------------|------------------------|
| 1 | 1 de 32 x 32 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 32 x 32 neuronas | – |
| | 32 de 32 x 32 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 16 x 16 neuronas | (Kernel = 2) 32 de 8x8 |
| 2 | 32 de 16 x 16 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 32 de 16 x 16 neuronas | – |
| | 64 de 16 x 16 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 8 x 8 neuronas | (Kernel = 1) 64 de 8x8 |
| 3 | 6144 neuronas | F.C.(DropOut : 0.5) | 3072 neuronas | – | 43 neuronas | – |

Modelos Propuestos

1 MODELO A

Perú

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|------------------------|---------------------|--------------------------------|----------|------------------------|--------------------------|
| 1 | 1 de 60 x 60 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 60 x 60 neuronas | – |
| | 32 de 60 x 60 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 30 x 30 neuronas | (Kernel = 2) 32 de 15x15 |
| 2 | 32 de 30 x 30 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 30 x 30 neuronas | – |
| | 64 de 30 x 30 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 15 x 15 neuronas | (Kernel = 1) 64 de 15x15 |
| 3 | 21600 neuronas | F.C.(DropOut : 0.5) | 3072 neuronas | – | 7 neuronas | – |

Modelos Propuestos

② MODELO B

Alemania

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|------------------------|---------------------|--------------------------------|----------|------------------------|-------------------------|
| 1 | 1 de 32 x 32 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 32 x 32 neuronas | – |
| | 32 de 32 x 32 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 16 x 16 neuronas | (Kernel = 4) 32 de 4x4 |
| 2 | 32 de 16 x 16 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 16 x 16 neuronas | – |
| | 64 de 16 x 16 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 8 x 8 neuronas | (Kernel = 2) 64 de 4x4 |
| 3 | 64 de 8 x 8 neuronas | Conv(DropOut : 0.6) | 128 de 5 x 5 | Activo | 128 de 8 x 8 neuronas | – |
| | 128 de 8 x 8 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 4 x 4 neuronas | (Kernel = 1) 128 de 4x4 |
| 4 | 3584 neuronas | F.C.(DropOut : 0.5) | 1024 neuronas | – | 43 neuronas | – |

Modelos Propuestos

② MODELO B

Perú

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|-------------------------|---------------------|--------------------------------|----------|-------------------------|-------------------------|
| 1 | 1 de 60 x 60 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 60 x 60 neuronas | – |
| | 32 de 60 x 60 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 30 x 30 neuronas | (Kernel = 4) 32 de 7x7 |
| 2 | 32 de 30 x 30 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 30 x 30 neuronas | – |
| | 64 de 30 x 30 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 15 x 15 neuronas | (Kernel = 2) 64 de 7x7 |
| 3 | 64 de 15 x 15 neuronas | Conv(DropOut : 0.6) | 128 de 5 x 5 | Activo | 128 de 15 x 15 neuronas | – |
| | 128 de 15 x 15 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 7 x 7 neuronas | (Kernel = 1) 128 de 7x7 |
| 4 | 10976 neuronas | F.C.(DropOut : 0.5) | 1024 neuronas | – | 7 neuronas | – |

Modelos Propuestos

3 MODELO C

Alemania

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|------------------------|---------------------|--------------------------------|----------|------------------------|-------------------------|
| 1 | 1 de 32 x 32 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 32 x 32 neuronas | – |
| | 32 de 32 x 32 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 16 x 16 neuronas | (Kernel = 4) 32 de 4x4 |
| 2 | 32 de 16 x 16 neuronas | Conv(DropOut : 0.7) | 64 de 3 x 3 | Activo | 64 de 16 x 16 neuronas | – |
| | 64 de 16 x 16 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 8 x 8 neuronas | (Kernel = 2) 64 de 4x4 |
| 3 | 64 de 8 x 8 neuronas | Conv(DropOut : 0.6) | 128 de 5 x 5 | Activo | 64 de 8 x 8 neuronas | – |
| | 128 de 8 x 8 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 4 x 4 neuronas | (Kernel = 1) 128 de 4x4 |
| 4 | 10976 neuronas | F.C.(DropOut : 0.5) | 1024 neuronas | – | 43 neuronas | – |

Modelos Propuestos

3 MODELO C

Perú

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|-------------------------|---------------------|--------------------------------|----------|-------------------------|-------------------------|
| 1 | 1 de 60 x 60 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 60 x 60 neuronas | – |
| | 32 de 60 x 60 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 30 x 30 neuronas | (Kernel = 4) 32 de 7x7 |
| 2 | 32 de 30 x 30 neuronas | Conv(DropOut : 0.7) | 64 de 3 x 3 | Activo | 64 de 30 x 30 neuronas | – |
| | 64 de 30 x 30 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 15 x 15 neuronas | (Kernel = 2) 64 de 7x7 |
| 3 | 64 de 15 x 15 neuronas | Conv(DropOut : 0.6) | 128 de 5 x 5 | Activo | 128 de 15 x 15 neuronas | – |
| | 128 de 15 x 15 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 7 x 7 neuronas | (Kernel = 1) 128 de 7x7 |
| 4 | 3584 neuronas | F.C.(DropOut : 0.5) | 1024 neuronas | – | 43 neuronas | – |

Modelos Propuestos

4 MODELO D

Alemania

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|------------------------|---------------------|--------------------------------|----------|------------------------|-------------------------|
| 1 | 1 de 32 x 32 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 32 x 32 neuronas | – |
| | 32 de 32 x 32 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 16 x 16 neuronas | (Kernel = 4) 32 de 4x4 |
| 2 | 32 de 16 x 16 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 16 x 16 neuronas | – |
| | 64 de 16 x 16 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 8 x 8 neuronas | (Kernel = 2) 64 de 4x4 |
| 3 | 64 de 8 x 8 neuronas | Conv(DropOut : 0.6) | 128 de 7 x 7 | Activo | 64 de 8 x 8 neuronas | – |
| | 128 de 8 x 8 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 4 x 4 neuronas | (Kernel = 1) 128 de 4x4 |
| 4 | 3584 neuronas | F.C.(DropOut : 0.5) | 1024 neuronas | – | 43 neuronas | – |

Modelos Propuestos

4 MODELO D

Perú

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|-------------------------|---------------------|--------------------------------|----------|-------------------------|-------------------------|
| 1 | 1 de 60 x 60 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 60 x 60 neuronas | – |
| | 32 de 60 x 60 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 30 x 30 neuronas | (Kernel = 4) 32 de 7x7 |
| 2 | 32 de 30 x 30 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 30 x 30 neuronas | – |
| | 64 de 30 x 30 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 15 x 15 neuronas | (Kernel = 2) 64 de 7x7 |
| 3 | 64 de 15 x 15 neuronas | Conv(DropOut : 0.6) | 128 de 7 x 7 | Activo | 128 de 15 x 15 neuronas | – |
| | 128 de 15 x 15 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 7 x 7 neuronas | (Kernel = 1) 128 de 7x7 |
| 4 | 10976 neuronas | F.C.(DropOut : 0.5) | 1024 neuronas | – | 7 neuronas | – |

Modelos Propuestos

5 MODELO E

Alemania

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|------------------------|---------------------|--------------------------------|----------|------------------------|-------------------------|
| 1 | 1 de 32 x 32 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 32 x 32 neuronas | – |
| | 32 de 32 x 32 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 16 x 16 neuronas | (Kernel = 8) 32 de 2x2 |
| 2 | 32 de 16 x 16 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 16 x 16 neuronas | – |
| | 64 de 16 x 16 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 8 x 8 neuronas | (Kernel = 4) 64 de 2x2 |
| 3 | 64 de 8 x 8 neuronas | Conv(DropOut : 0.6) | 128 de 5 x 5 | Activo | 128 de 8 x 8 neuronas | – |
| | 128 de 8 x 8 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 4 x 4 neuronas | (Kernel = 2) 128 de 2x2 |
| 4 | 128 de 4 x 4 neuronas | Conv(DropOut : 0.6) | 128 de 7 x 7 | Activo | 128 de 4 x 4 neuronas | – |
| | 128 de 4 x 4 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 2 x 2 neuronas | (Kernel = 1) 128 de 2x2 |
| 5 | 1408 neuronas | F.C.(DropOut : 0.5) | 702 neuronas | – | 43 neuronas | – |

Modelos Propuestos

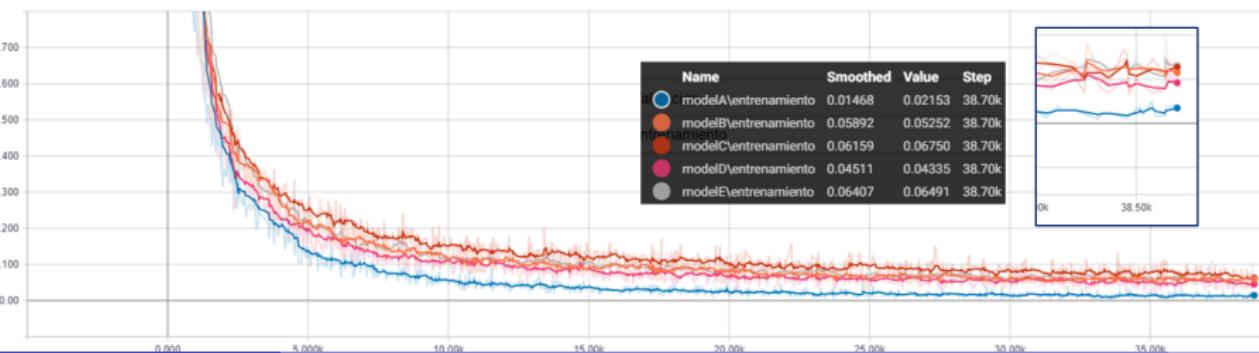
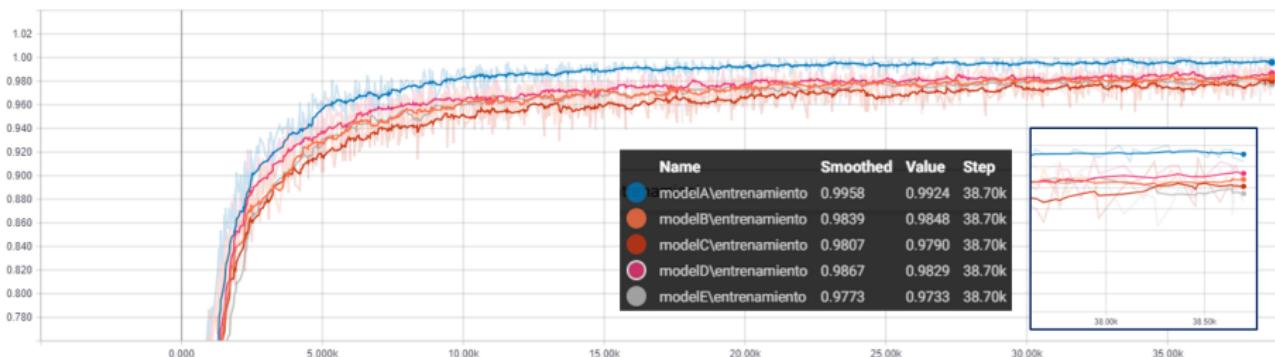
5 MODELO E

Perú

| Capa | Entrada | Tipo | Número de (kernels/filtros) | Padding | Salida | Func.Esc.Múltiple |
|------|-------------------------|---------------------|--------------------------------|----------|-------------------------|-------------------------|
| 1 | 1 de 60 x 60 neuronas | Conv(DropOut : 0.8) | 32 de 3 x 3 | Activo | 32 de 60 x 60 neuronas | – |
| | 32 de 60 x 60 neuronas | Max Pool | 32 de 2 x 2 | Inactivo | 32 de 30 x 30 neuronas | (Kernel = 8) 32 de 3x3 |
| 2 | 32 de 30 x 30 neuronas | Conv(DropOut : 0.7) | 64 de 5 x 5 | Activo | 64 de 30 x 30 neuronas | – |
| | 64 de 30 x 30 neuronas | Max Pool | 64 de 2 x 2 | Inactivo | 64 de 15 x 15 neuronas | (Kernel = 4) 64 de 3x3 |
| 3 | 64 de 15 x 15 neuronas | Conv(DropOut : 0.6) | 128 de 5 x 5 | Activo | 128 de 15 x 15 neuronas | – |
| | 128 de 15 x 15 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 7 x 7 neuronas | (Kernel = 2) 128 de 3x3 |
| 4 | 128 de 7 x 7 neuronas | Conv(DropOut : 0.6) | 128 de 7 x 7 | Activo | 128 de 7 x 7 neuronas | – |
| | 128 de 7 x 7 neuronas | Max Pool | 128 de 2 x 2 | Inactivo | 128 de 3 x 3 neuronas | (Kernel = 1) 128 de 3x3 |
| 5 | 3168 neuronas | F.C.(DropOut : 0.5) | 702 neuronas | – | 43 neuronas | – |

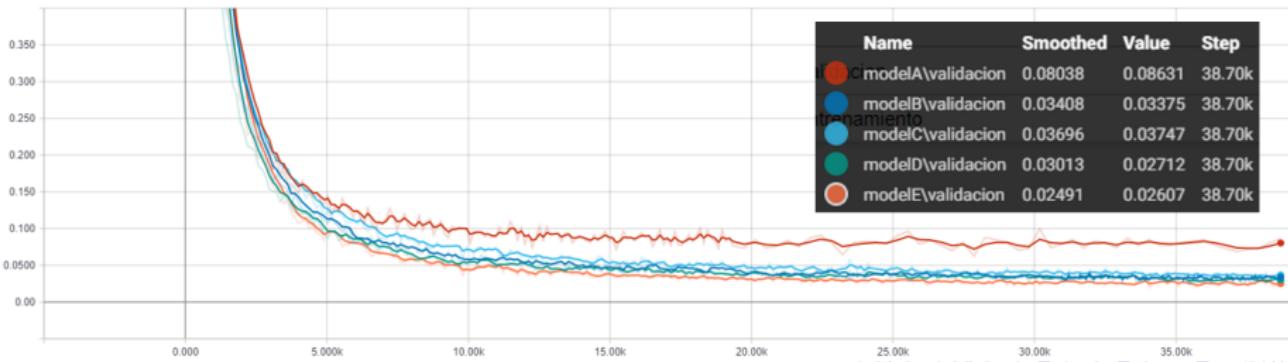
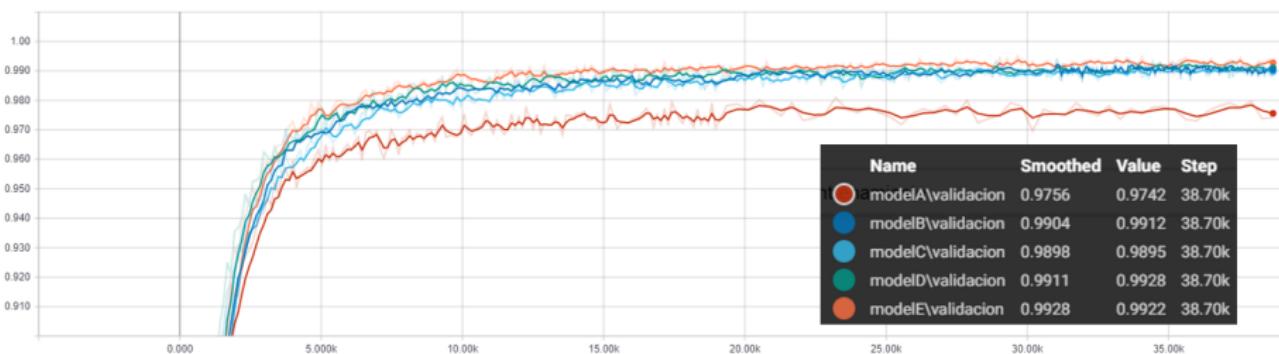
Resultados del Entrenamiento - Dataset Alemania

Los 5 modelos lograron similares resultados de acierto y error durante el entrenamiento.



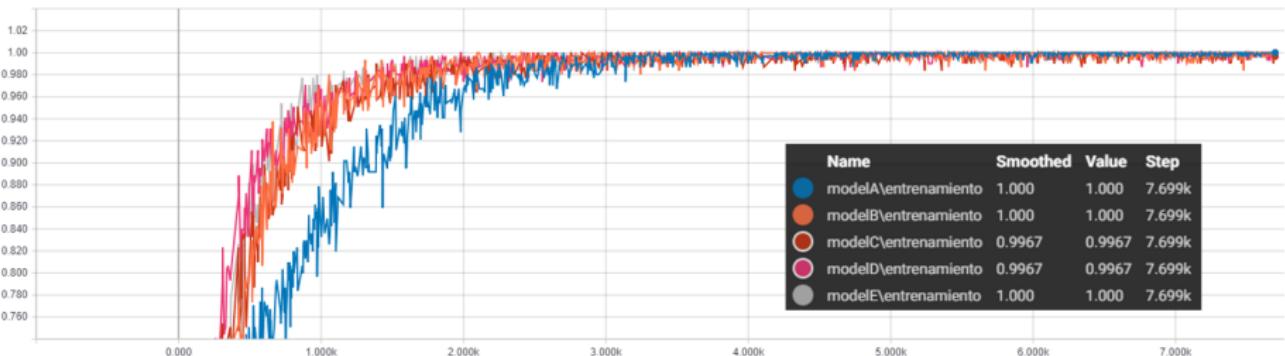
Resultados de Validación - Dataset Alemania

El Modelo E obtuvo una de las mejores tasas de Acierto y Error.



Resultados del Entrenamiento - Dataset Perú

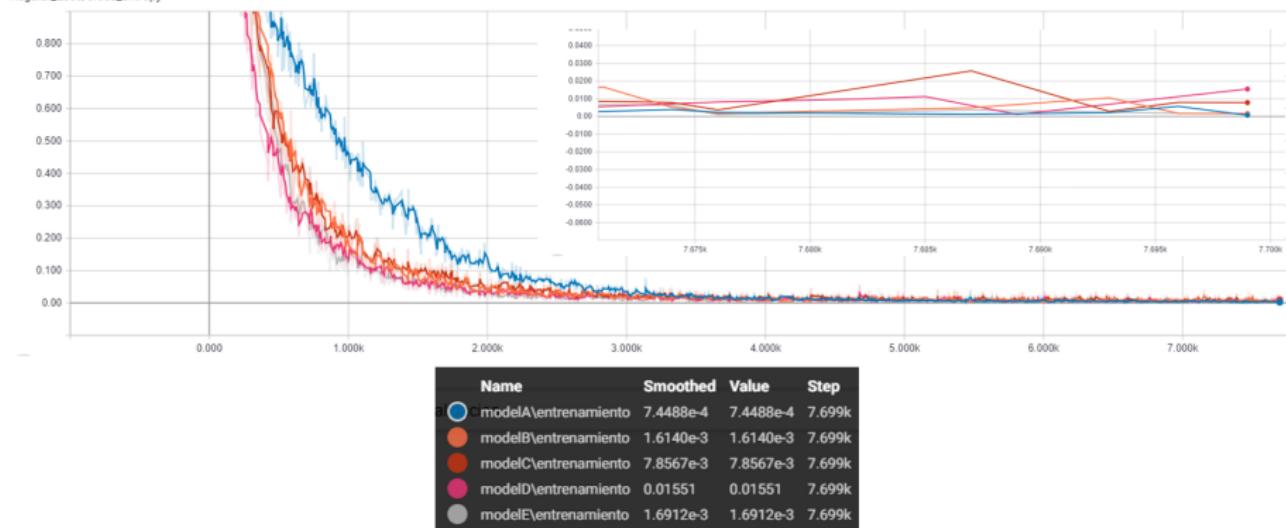
Los 5 modelos lograron similares resultados de acierto y error durante el entrenamiento.



Resultados del Entrenamiento - Dataset Perú

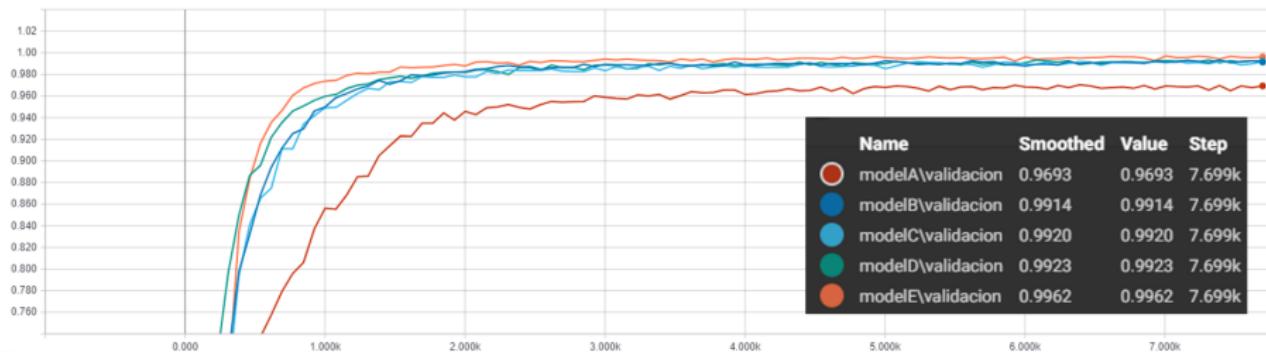
Los 5 modelos lograron similares resultados de acierto y error durante el entrenamiento.

Regular_Loss/cross_entropy



Resultados de Validación - Dataset Perú

El Modelo E obtuvo una de las mejores tasas de Acierto y Error.



Regular_Loss/cross_entropy

| Name | Smoothed | Value | Step |
|-------------------|----------|---------|--------|
| modelA\validacion | 0.1155 | 0.1155 | 7.699k |
| modelB\validacion | 0.02733 | 0.02733 | 7.699k |
| modelC\validacion | 0.02610 | 0.02610 | 7.699k |
| modelD\validacion | 0.02587 | 0.02587 | 7.699k |
| modelE\validacion | 0.01841 | 0.01841 | 7.699k |

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 **Resultados de la tesis**
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Resultados de la tesis

Al culminar con la investigación se llegaron a resultados interesantes del punto de vista tanto teórico como computacional. ...

- **Teóricos:**

Resultados de la tesis

Al culminar con la investigación se llegaron a resultados interesantes del punto de vista tanto teórico como computacional. ...

- **Teóricos:**



Resultados de la tesis

Al culminar con la investigación se llegaron a resultados interesantes del punto de vista tanto teórico como computacional. ...

- **Teóricos:**

- ▶
- ▶

- **Computacionales:**

- **Computacionales:**



- **Computacionales:**

- ▶
- ▶

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Consideraciones finales

- **Conclusiones:**

Consideraciones finales

- **Conclusiones:**



Consideraciones finales

- **Conclusiones:**

- ▶
- ▶

- **Trabajos futuros:**

- **Trabajos futuros:**



- **Trabajos futuros:**

- ▶
- ▶

Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Validación Cruzada
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas

Referencias bibliograficas

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- Dong, X. and Zhou, D.-X. (2008). Learning gradients by a gradient descent algorithm. *Journal of Mathematical Analysis and Applications*, 341(2):1018 – 1027.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA. ACM.
- Jordan, J. (2018). Setting the learning rate of your neural network. [Online; accessed Agosto 08, 2018].
- Karpathy, A. (2016). Convolutional neural networks (cnns / convnets). *Cs231-Stanford Course*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks.