Redes Neuronales y DL

Alvaro J. Riascos Villegas

Junio de 2019

Contenido

Redes Neuronales

- La redes neuronales son un modelo de función de aprendizaje motivado por las conexiones entre neuronas en el cerebro.
- El prototipo de modelo es la red neuronal (feedforward) o perceptor multicapas.
- Uno de los principales problemas que surgen es que la optimización (de la función de verosimilitud) deja de ser convexa y pueden existir múltiples puntos críticos locales.
- La maximización de la función de verosimilitud o minimización del error cuadrático requiere del cálculo de algunas derivadas. Esto se puede hacer eficientemente usando la técnica de propagación hacia atrás (*backpropagation*).

Redes Neuronales

- La redes neuronales son un modelo de función de aprendizaje motivado por las conexiones entre neuronas en el cerebro.
- El prototipo de modelo es la red neuronal (feedforward) o perceptor multicapas.
- Uno de los principales problemas que surgen es que la optimización (de la función de verosimilitud) deja de ser convexa y pueden existir múltiples puntos críticos locales.
- La maximización de la función de verosimilitud o minimización del error cuadrático requiere del cálculo de algunas derivadas. Esto se puede hacer eficientemente usando la técnica de propagación hacia atrás (*backpropagation*).

Redes Neuronales

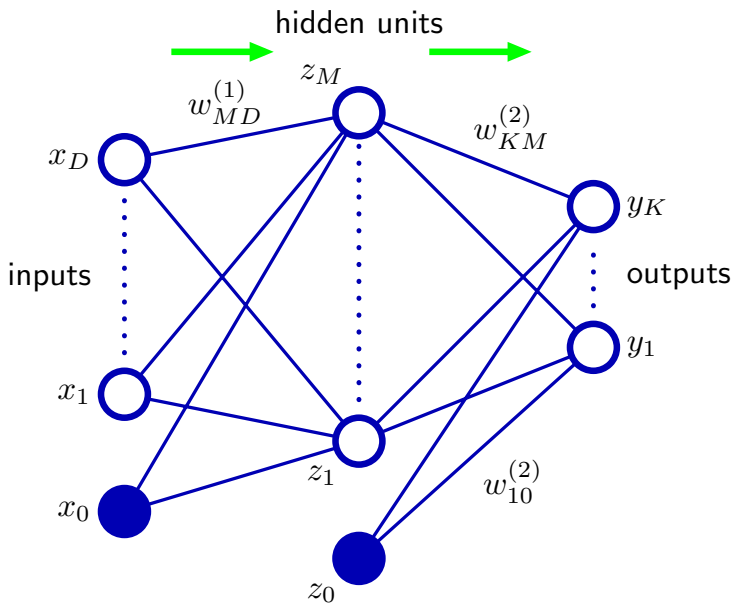
- La redes neuronales son un modelo de función de aprendizaje motivado por las conexiones entre neuronas en el cerebro.
- El prototipo de modelo es la red neuronal (feedforward) o perceptor multicapas.
- Uno de los principales problemas que surgen es que la optimización (de la función de verosimilitud) deja de ser convexa y pueden existir múltiples puntos críticos locales.
- La maximización de la función de verosimilitud o minimización del error cuadrático requiere del cálculo de algunas derivadas. Esto se puede hacer eficientemente usando la técnica de propagación hacia atrás (*backpropagation*).

Redes Neuronales

- La redes neuronales son un modelo de función de aprendizaje motivado por las conexiones entre neuronas en el cerebro.
- El prototipo de modelo es la red neuronal (feedforward) o perceptor multicapas.
- Uno de los principales problemas que surgen es que la optimización (de la función de verosimilitud) deja de ser convexa y pueden existir múltiples puntos críticos locales.
- La maximización de la función de verosimilitud o minimización del error cuadrático requiere del cálculo de algunas derivadas. Esto se puede hacer eficientemente usando la técnica de propagación hacia atrás (*backpropagation*).

Redes Neuronales

- Por simplicidad vamos a describir los detalles de una red neuronal con dos capas.



Redes Neuronales

- Supongamos que tenemos D variables de entrada: $\{x_1, ..., x_D\}$ y M variables de salida $\{y_1, ..., y_M\}$.
- Sean h_1, h_2 las funciones de activación (h_2 la función de activación de las salidas de la red).
- Entonces las siguientes ecuaciones definen una red neuronal (feed-forward):

Redes Neuronales

- Supongamos que tenemos D variables de entrada: $\{x_1, ..., x_D\}$ y M variables de salida $\{y_1, ..., y_M\}$.
- Sean h_1, h_2 las funciones de activación (h_2 la función de activación de las salidas de la red).
- Entonces las siguientes ecuaciones definen una red neuronal (feed-forward):

Redes Neuronales

- Supongamos que tenemos D variables de entrada: $\{x_1, ..., x_D\}$ y M variables de salida $\{y_1, ..., y_M\}$.
- Sean h_1, h_2 las funciones de activación (h_2 la función de activación de las salidas de la red).
- Entonces las siguientes ecuaciones definen una red neuronal (feed-forward):

$$a_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

$$z_j^{(1)} = h_1(a_j^{(1)})$$

$$a_j^{(2)} = \sum_{i=1}^{M} w_{ji}^{(2)} z_i^{(1)} + w_{j0}^{(2)}$$

$$z_j^{(2)} = h_2(a_j^{(2)})$$

donde $w_{j0}^{(1)}, w_{j0}^{(2)}$ representan los sesgos en cada capa.

- Definamos $z_i^{(0)} = x_i, y_j = z_j^{(2)}$.
- Si definimos variables adicionales $x_0 = 1, z_0^{(1)} = 1$ y $z_0^{(2)} = 1$, podemos escribir la red como:

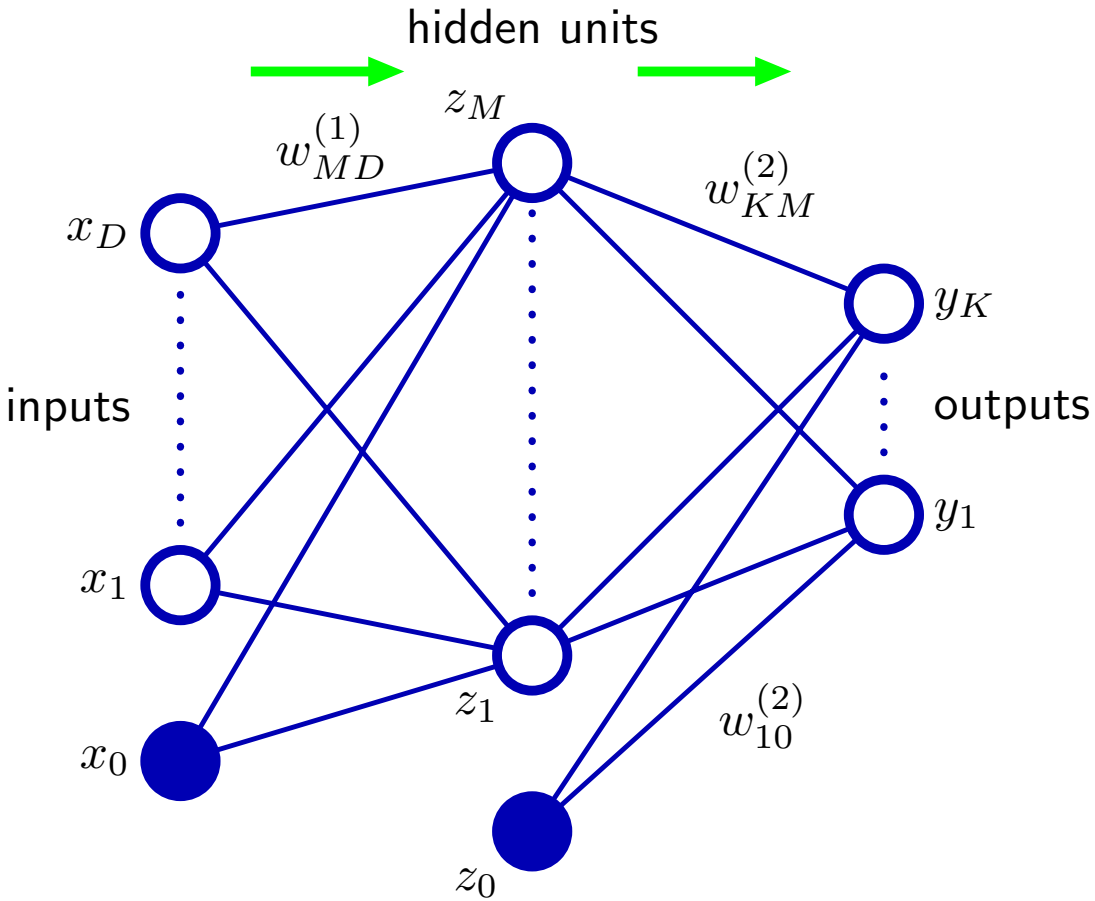
$$y_k(x, w) = h_2(\sum_{j=0}^{M} w_{kj}^{(2)} h_1(\sum_{i=0}^{D} w_{ji}^{(1)} x_i))$$

- Definamos $z_i^{(0)} = x_i, y_j = z_j^{(2)}$.
- Si definimos variables adicionales $x_0 = 1, z_0^{(1)} = 1$ y $z_0^{(2)} = 1$, podemos escribir la red como:

$$y_k(x, w) = h_2(\sum_{j=0}^{M} w_{kj}^{(2)} h_1(\sum_{i=0}^{D} w_{ji}^{(1)} x_i))$$

Red neuronal con dos capas

La terminología refleja el hecho de que es necesario estimar dos conjunto de pesos.



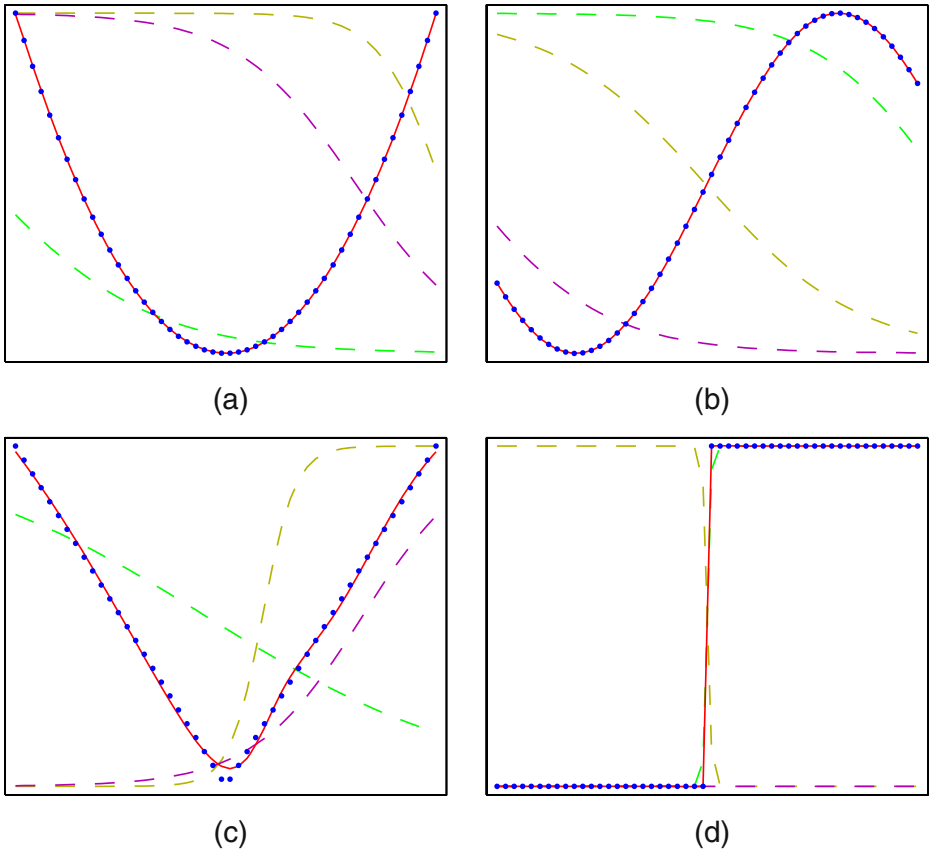
Red neuronal con dos capas

- Propiedad de aproximación universal: Una red neuronal de dos capas y salidas lineales, puede aproximar uniformemente a cualquier función continua con dominio compacto siempre y cuando se tengan suficientes neuronas.
- Este resultado es válido para muchos tipos de funciones de activación en la unidades ocultas (excepto polinomios).

Red neuronal con dos capas

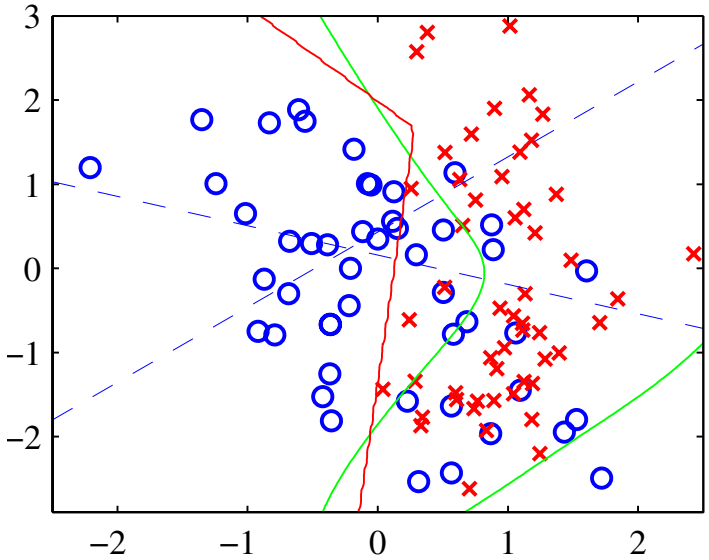
- Propiedad de aproximación universal: Una red neuronal de dos capas y salidas lineales, puede aproximar uniformemente a cualquier función continua con dominio compacto siempre y cuando se tengan suficientes neuronas.
- Este resultado es válido para muchos tipos de funciones de activación en la unidades ocultas (excepto polinomios).

Red neuronal con dos capas, tres neuronas (regresión)



- Capacidad de aproximación de una red. Los datos son los 50 puntos azules. Se entrena una red con dos capas, tres neuronas, función de activación tanh, salidas lineales. Las salidas de las tres neuronas ocultas se muestran con líneas punteadas.

Red neuronal con dos capas, dos neuronas (clasificación)



- Capacidad de aproximación de una red: las lineas punteadas son las salidas de cada una de las dos neuronas (hipersuperifices). Funciones de activación tanh y salida logistica sigmoid.
- La línea verde es el clasificador Bayesiano. La roja el clasificador de la red.

Contenido

Interpretación probabilística, funciones de error y activación

- Vamos a considerar tres casos que nos permitirán tener cierta intuición sobre qué funciones de activiación y función de error son adecuadas para utilizar en la redes.
- Esto depende del tipo de problema:
  1. Problemas de regresión: salidas lineales y sumas de errores cuadráticos.
  2. Clasificación binaria: salidas logísticas sigmoid y error entropía cruzada.
  3. Para variables categoricas: salidas softmax y error la generalizacion de la entropia cruzada a multiples clases. Esto ofrece una alternativa para el caso de clasificación binaria.

Interpretación probabilística, funciones de error y activación

- Vamos a considerar tres casos que nos permitirán tener cierta intuición sobre qué funciones de activiación y función de error son adecuadas para utilizar en la redes.
- Esto depende del tipo de problema:
  1. Problemas de regresión: salidas lineales y sumas de errores cuadráticos.
  2. Clasificación binaria: salidas logísticas sigmoid y error entropía cruzada.
  3. Para variables categoricas: salidas softmax y error la generalizacion de la entropia cruzada a multiples clases. Esto ofrece una alternativa para el caso de clasificación binaria.

Problema de regresión: una variable de salida

- Supongamos que la muestra $\tau = \{(x_1, t_1), ..., (x_N, t_N)\}$ es i.i.d. y que la distribución condicional $p(t \mid x, w, \beta)$ de t es de la forma:

$$p(t \mid x, w, \beta) = N(t \mid y(x, w), \beta^{-1})$$

- Es decir, la distribución condicional de las salidas a las entradas es una distribción normal con una media que depende de la salida de la red $y(x, w)$ y β es el parámetro de precisión.

Problema de regresión: una variable de salida

- En este caso la función de verosimilitud es:

$$L(t, X, w, \beta) = \prod p(t_n \mid x_n, w, \beta)$$

- Si escribimos la función negativa de la función log verosimilitud obtenemos:

$$\frac{\beta}{2}\sum_{n=1}^{N}(y(x, w) - t_n)^2 - \frac{N}{2}\log(\beta) + \frac{N}{2}\log(2\pi)$$

- Luego la maximización de la función de verosimilitud es equivalente a minimizar:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$$

y β satisface:

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}(y(x_n, \widehat{w}) - t_n)^2$$

Problema de regresión: una variable de salida

- En este caso la función de verosimilitud es:

$$L(t, X, w, \beta) = \prod p(t_n \mid x_n, w, \beta)$$

- Si escribimos la función negativa de la función log verosimilitud obtenemos:

$$\frac{\beta}{2}\sum_{n=1}^{N}(y(x, w) - t_n)^2 - \frac{N}{2}\log(\beta) + \frac{N}{2}\log(2\pi)$$

- Luego la maximización de la función de verosimilitud es equivalente a minimizar:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$$

y β satisface:

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}(y(x_n, \widehat{w}) - t_n)^2$$

Problema de regresión: una variable de salida

- En este caso la función de verosimilitud es:

$$L(t, X, w, \beta) = \prod p(t_n \mid x_n, w, \beta)$$

- Si escribimos la función negativa de la función log verosimilitud obtenemos:

$$\frac{\beta}{2}\sum_{n=1}^{N}(y(x, w) - t_n)^2 - \frac{N}{2}\log(\beta) + \frac{N}{2}\log(2\pi)$$

- Luego la maximización de la función de verosimilitud es equivalente a minimizar:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$$

y β satisface:

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}(y(x_n, \widehat{w}) - t_n)^2$$

Problema de regresión: una variable de salida

- En este caso la función de verosimilitud es:

$$L(t, X, w, \beta) = \prod p(t_n \mid x_n, w, \beta)$$

- Si escribimos la función negativa de la función log verosimilitud obtenemos:

$$\frac{\beta}{2}\sum_{n=1}^{N}(y(x, w) - t_n)^2 - \frac{N}{2}\log(\beta) + \frac{N}{2}\log(2\pi)$$

- Luego la maximización de la función de verosimilitud es equivalente a minimizar:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$$

y β satisface:

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}(y(x_n, \widehat{w}) - t_n)^2$$

Problema de regresión: una variable de salida

- En este caso la función de verosimilitud es:

$$L(t, X, w, \beta) = \prod p(t_n \mid x_n, w, \beta)$$

- Si escribimos la función negativa de la función log verosimilitud obtenemos:

$$\frac{\beta}{2}\sum_{n=1}^{N}(y(x, w) - t_n)^2 - \frac{N}{2}\log(\beta) + \frac{N}{2}\log(2\pi)$$

- Luego la maximización de la función de verosimilitud es equivalente a minimizar:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$$

y β satisface:

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}(y(x_n, \widehat{w}) - t_n)^2$$

Problema de clasificación binario

- Supongamos ahora que usamos como función de activación de salida la función logística: $y = \frac{1}{1+\exp(-a)}$.
- La distribución condicional $p(t \mid x, w)$ es:

$$p(t \mid x, w) = y(x, w)^t (1 - y(x, w))^{1-t}$$

- Suponiendo que la muestra es i.i.d entonces la función negativa de la función de verosimilitud es:

$$E(w) = -\sum_{n=1}^{N} (t_n \log(y_n) + (1 - t_n)(1 - \log(y_n)))$$

donde $y_n = y(x_n, w)$. Esta función es también conocida como entropia cruzada.

Problema de clasificación binario

- Supongamos ahora que usamos como función de activación de salida la función logística: $y = \frac{1}{1+\exp(-a)}$.
- La distribución condicional $p(t \mid x, w)$ es:

$$p(t \mid x, w) = y(x, w)^t (1 - y(x, w))^{1-t}$$

- Suponiendo que la muestra es i.i.d entonces la función negativa de la función de verosimilitud es:

$$E(w) = -\sum_{n=1}^{N} (t_n \log(y_n) + (1 - t_n)(1 - \log(y_n)))$$

donde $y_n = y(x_n, w)$. Esta función es también conocida como entropia cruzada.

Problema de clasificación binario

- Supongamos ahora que usamos como función de activación de salida la función logística: $y = \frac{1}{1+\exp(-a)}$.
- La distribución condicional $p(t \mid x, w)$ es:

$$p(t \mid x, w) = y(x, w)^t (1 - y(x, w))^{1-t}$$

- Suponiendo que la muestra es i.i.d entonces la función negativa de la función de verosimilitud es:

$$E(w) = -\sum_{n=1}^{N} (t_n \log(y_n) + (1 - t_n)(1 - \log(y_n)))$$

donde $y_n = y(x_n, w)$. Esta función es también conocida como entropia cruzada.

Problema de clasificación binario

- En este caso no existe un análogo a la precisión (del caso de regresión) pero se puede extender el modelo para permitir errores en la marca.
- Usar la la entropia cruzada es más eficiente computacionalmente que usar error cuadrático y generaliza mejor (Simard et.al (2003)).
- Todo generaliza de forma natural al caso de **varias clasificaciones binarias independientes**.

Problema de clasificación binario

- En este caso no existe un análogo a la precisión (del caso de regresión) pero se puede extender el modelo para permitir errores en la marca.
- Usar la la entropia cruzada es más eficiente computacionalmente que usar error cuadrático y generaliza mejor (Simard et.al (2003)).
- Todo generaliza de forma natural al caso de **varias clasificaciones binarias independientes**.

Problema de clasificación binario

- En este caso no existe un análogo a la precisión (del caso de regresión) pero se puede extender el modelo para permitir errores en la marca.
- Usar la la entropia cruzada es más eficiente computacionalmente que usar error cuadrático y generaliza mejor (Simard et.al (2003)).
- Todo generaliza de forma natural al caso de **varias clasificaciones binarias independientes**.

Múltiples problemas de clasificación binarios (independientes)

- La distribución condicional $p(t \mid x, w)$ es:

$$p(t \mid x, w) = y_k(x, w)^{t_k}(1 - y_k(x, w))^{1-t_k}$$

$k = 1, ..., K$

- Suponiendo que la muestra es i.i.d entonces la función negativa de la función de verosimilitud es:

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K}(t_{nk}\log(y_{nk}) + (1 - t_{nk})(1 - \log(y_{nk})))$$

donde $y_{nk} = y_k(x_n, w)$

Múltiples problemas de clasificación binarios (independientes)

- La distribución condicional $p(t \mid x, w)$ es:

$$p(t \mid x, w) = y_k(x, w)^{t_k}(1 - y_k(x, w))^{1-t_k}$$

$k = 1, ..., K$

- Suponiendo que la muestra es i.i.d entonces la función negativa de la función de verosimilitud es:

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K}(t_{nk}\log(y_{nk}) + (1 - t_{nk})(1 - \log(y_{nk})))$$

donde $y_{nk} = y_k(x_n, w)$

Problema de clasificación en varias categorias

- Supongamos que una instancia puede esta en una de K categorias mutuamente excluyentes.
- La función de error en este caso es:

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K}(t_{kn}\log(y_k(x_n, w))$$

y la función de activación de salida es:

$$y_k(x, w) = \frac{\exp(a_k(x, w)}{\sum_j \exp(a_j(x, w))}$$

Problema de clasificación en varias categorias

- Supongamos que una instancia puede esta en una de K categorias mutuamente excluyentes.
- La función de error en este caso es:

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K}(t_{kn}\log(y_k(x_n, w))$$

y la función de activación de salida es:

$$y_k(x, w) = \frac{\exp(a_k(x, w)}{\sum_j \exp(a_j(x, w))}$$

Contenido

Optimización de los parámetros

- Las redes neuronales tiene una multiplicidad de representaciones derivadas de las simetría con respecto a los parámetros. Esto explica que existan muchos mínimos equivalentes.
- También pueden existir mínimos locales.
- Los métodos numéricos todos se basan de algun forma en una iteración de la forma:

$$w^{\tau+1} = w^{\tau} + \Delta w^{\tau}$$

Optimización de los parámetros

- Las redes neuronales tiene una multiplicidad de representaciones derivadas de las simetría con respecto a los parámetros. Esto explica que existan muchos mínimos equivalentes.
- También pueden existir mínimos locales.
- Los métodos numéricos todos se basan de algun forma en una iteración de la forma:

$$w^{\tau+1} = w^{\tau} + \Delta w^{\tau}$$

Gradiente descendente

- El error, en un punto crítico w^* puede aproximarse localmente por:

$$E(w) \approx E(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$

y una estrategia para aproximarse al punto crítico es usar la información del gradiente. La implementación más sencilla de esta idea es el método del gradiente descendente:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E(w^{\tau})$$

donde η se conoce como la tasa de aprendizaje.

- Obsérvese que para calcular $\bigtriangledown w^{\tau}$ es necesario usar toda la muestra de entrenamiento (i.e., se conoce como método batch).

Gradiente descendente

- El error, en un punto crítico w^* puede aproximarse localmente por:

$$E(w) \approx E(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$

y una estrategia para aproximarse al punto crítico es usar la información del gradiente. La implementación más sencilla de esta idea es el método del gradiente descendente:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E(w^{\tau})$$

donde η se conoce como la tasa de aprendizaje.

- Obsérvese que para calcular $\bigtriangledown w^{\tau}$ es necesario usar toda la muestra de entrenamiento (i.e., se conoce como método batch).

Gradiente descendente

- El error, en un punto crítico w^* puede aproximarse localmente por:

$$E(w) \approx E(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$

y una estrategia para aproximarse al punto crítico es usar la información del gradiente. La implementación más sencilla de esta idea es el método del gradiente descendente:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E(w^{\tau})$$

donde η se conoce como la tasa de aprendizaje.

- Obsérvese que para calcular $\bigtriangledown w^{\tau}$ es necesario usar toda la muestra de entrenamiento (i.e., se conoce como método batch).

Gradiente descendente secuencial o estocástico

- El error en muchos problemas es de la forma: $E(w) = \sum_{i=1}^{N} E_n(w)$, donde $E_n(w)$ depende únicamente del error que se tiene con el ejemplo (x_n, t_n).
- La idea del método del gradiente descendente es iterar estocásticamente sobre n en la ecuación:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E_n(w^{\tau})$$

- Obsérvese que en cada actualización solo se usa un punto de la base de entrenamiento elegido aleatoriamente.
- Se conoce como un método en línea (online).
- Alternativamente se puede elegir un muestra pequeña (mini batch) para calcular el $\bigtriangledown E_{n_i}(w^{\tau})$.
- Una tercera alternativa es iterar secuencialmente sobre la muestra (elegir aleatoriamente puntos sin reemplazo de la muestra).

Gradiente descendente secuencial o estocástico

- El error en muchos problemas es de la forma: $E(w) = \sum_{i=1}^{N} E_n(w)$, donde $E_n(w)$ depende únicamente del error que se tiene con el ejemplo (x_n, t_n).
- La idea del método del gradiente descendente es iterar estocásticamente sobre n en la ecuación:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E_n(w^{\tau})$$

- Obsérvese que en cada actualización solo se usa un punto de la base de entrenamiento elegido aleatoriamente.
- Se conoce como un método en línea (online).
- Alternativamente se puede elegir un muestra pequeña (mini batch) para calcular el $\bigtriangledown E_{n_i}(w^{\tau})$.
- Una tercera alternativa es iterar secuencialmente sobre la muestra (elegir aleatoriamente puntos sin reemplazo de la muestra).

Gradiente descendente secuencial o estocástico

- El error en muchos problemas es de la forma: $E(w) = \sum_{i=1}^{N} E_n(w)$, donde $E_n(w)$ depende únicamente del error que se tiene con el ejemplo (x_n, t_n).
- La idea del método del gradiente descendente es iterar estocásticamente sobre n en la ecuación:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E_n(w^{\tau})$$

- Obsérvese que en cada actualización solo se usa un punto de la base de entrenamiento elegido aleatoriamente.
- Se conoce como un método en línea (online).
- Alternativamente se puede elegir un muestra pequeña (mini batch) para calcular el $\bigtriangledown E_{n_i}(w^{\tau})$.
- Una tercera alternativa es iterar secuencialmente sobre la muestra (elegir aleatoriamente puntos sin reemplazo de la muestra).

Gradiente descendente secuencial o estocástico

- El error en muchos problemas es de la forma: $E(w) = \sum_{i=1}^{N} E_n(w)$, donde $E_n(w)$ depende únicamente del error que se tiene con el ejemplo (x_n, t_n).
- La idea del método del gradiente descendente es iterar estocásticamente sobre n en la ecuación:

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E_n(w^{\tau})$$

- Obsérvese que en cada actualización solo se usa un punto de la base de entrenamiento elegido aleatoriamente.
- Se conoce como un método en línea (online).
- Alternativamente se puede elegir un muestra pequeña (mini batch) para calcular el $\bigtriangledown E_{n_i}(w^{\tau})$.
- Una tercera alternativa es iterar secuencialmente sobre la muestra (elegir aleatoriamente puntos sin reemplazo de la muestra).

Gradiente descendente secuencial o estocástico

- Es un método útil porque evita calculos redundantes (e.g., suponga que se hacen dos copias de la base de datos).

Back propagation

- El método más popular de entrenamiento de una red es conocido como back propagation y es una forma eficiente de estimar $\bigtriangledown E$.
- Dada una función de error: $E(w) = \sum_{i=1}^{N} E_n(w)$ vamos a concentrarnos en estimar el gradiente de E_n.

Back propagation

Aprovechando la estrutura recursiva del modelo se puede demostrar que:

1. Deltas:

$$\delta^{[L]} = h'(a^{[L]}) \cdot (z^L - y)$$

$$\delta^{[l]} = h(z^{[l]}) \cdot (W^{[l+1]})^T \delta^{[l+1]}$$

2. Derivadas parciales:

$$\frac{\partial E}{\partial b_j^{[l]}} = \delta_j^{[l]}$$

$$\frac{\partial E}{\partial w_{jk}^{[l]}} = z_k^{[l-1]}$$

- Osérvese que se pueden estimar todos los z y a haciendo una simulación hacia adelante del modelo (forward pass). Con esto valores se obtiene los *delta*.
- Con una simulación hacia atrás se obtienen las derivadas parciales del error con respecto a los parámetros.

Contenido

Estimación de la Pobreza usando Aprendizaje de Máquinas

- Muchos países no tienen estimaciones confiables de la riqueza (activos) o consumo.
- Es es un problema más pronunciado a nivel local.
- Basado en: Combining satellite imagery and machine learning to predict poverty. 2016. Science.
- La estrategia del artículo es:
  1. Pre-entrenar una red neuronal convolucional.
  2. Estimar una red convolucional para predecir la luminosidad de una región usando imagenes satelitales. Esta red permite caracterizar en baja dimensión los aspectos relevantes de las imágenes que caracterizan la luminosidad (features).
  3. Entrenar para países (comparables), con mediciones de pobreza confiables, un modelo de aprendizaje de máqinas que (regresión de Ridge), usando los features de la red del segundo paso, caracterice medidas de riqueza (activos) y consumo.

Estimación de la Pobreza usando Aprendizaje de Máquinas

- Muchos países no tienen estimaciones confiables de la riqueza (activos) o consumo.
- Es es un problema más pronunciado a nivel local.
- Basado en: Combining satellite imagery and machine learning to predict poverty. 2016. Science.
- La estrategia del artículo es:
  1. Pre-entrenar una red neuronal convolucional.
  2. Estimar una red convolucional para predecir la luminosidad de una región usando imagenes satelitales. Esta red permite caracterizar en baja dimensión los aspectos relevantes de las imágenes que caracterizan la luminosidad (features).
  3. Entrenar para países (comparables), con mediciones de pobreza confiables, un modelo de aprendizaje de máqinas que (regresión de Ridge), usando los features de la red del segundo paso, caracterice medidas de riqueza (activos) y consumo.

Estimación de la Pobreza usando Aprendizaje de Máquinas

- Muchos países no tienen estimaciones confiables de la riqueza (activos) o consumo.
- Es es un problema más pronunciado a nivel local.
- Basado en: Combining satellite imagery and machine learning to predict poverty. 2016. Science.
- La estrategia del artículo es:
  1. Pre-entrenar una red neuronal convolucional.
  2. Estimar una red convolucional para predecir la luminosidad de una región usando imagenes satelitales. Esta red permite caracterizar en baja dimensión los aspectos relevantes de las imágenes que caracterizan la luminosidad (features).
  3. Entrenar para países (comparables), con mediciones de pobreza confiables, un modelo de aprendizaje de máqinas que (regresión de Ridge), usando los features de la red del segundo paso, caracterice medidas de riqueza (activos) y consumo.

Estimación de la Pobreza usando Aprendizaje de Máquinas

- Muchos países no tienen estimaciones confiables de la riqueza (activos) o consumo.
- Es es un problema más pronunciado a nivel local.
- Basado en: Combining satellite imagery and machine learning to predict poverty. 2016. Science.
- La estrategia del artículo es:
  1. Pre-entrenar una red neuronal convolucional.
  2. Estimar una red convolucional para predecir la luminosidad de una región usando imagenes satelitales. Esta red permite caracterizar en baja dimensión los aspectos relevantes de las imágenes que caracterizan la luminosidad (features).
  3. Entrenar para países (comparables), con mediciones de pobreza confiables, un modelo de aprendizaje de máqinas que (regresión de Ridge), usando los features de la red del segundo paso, caracterice medidas de riqueza (activos) y consumo.

Motivación

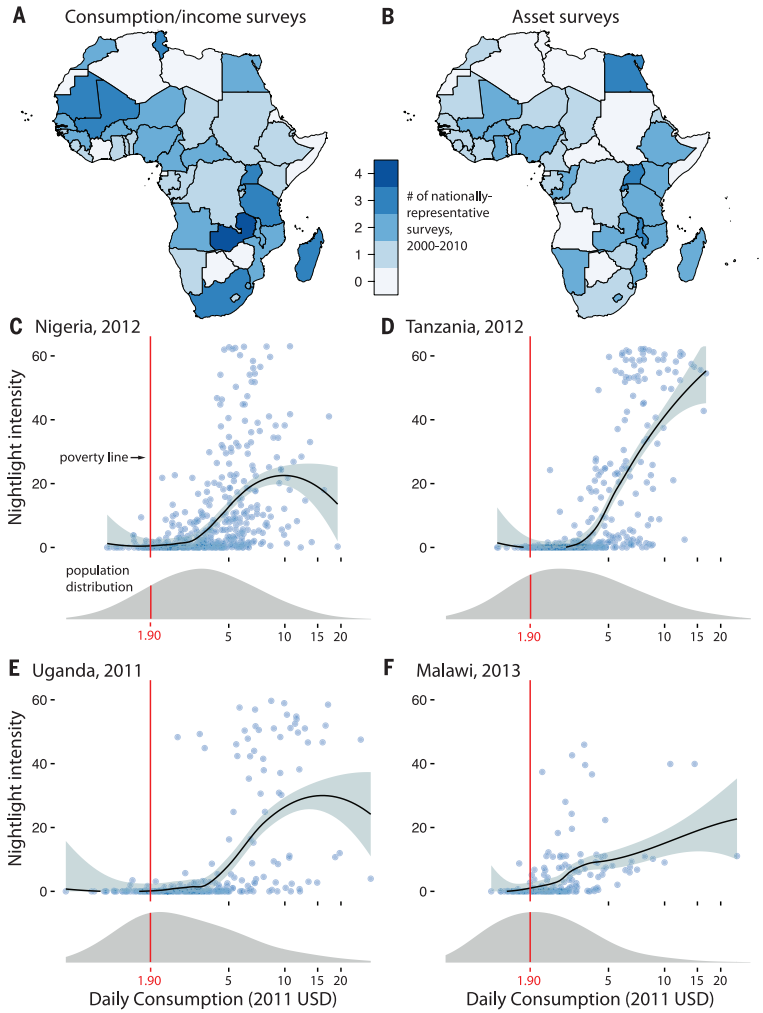


Fig. 1. Poverty data gaps. (**A**) Number of nationally representative consumption surveys occurring in each African country between 2000 and 2010. (**B**) Same as (A), for DHS surveys measuring assets. (**C** to **F**) Relationship between per capita consumption expenditure (measured in U.S. dollars) and nightlight intensity at the cluster level for four African countries, based on household surveys. Nationally representative share of households at each point in the consumption distribution is shown beneath each panel in gray. Vertical red lines show the official international extreme poverty line ($1.90 per person per day), and black lines are fits to the data with corresponding 95% confidence intervals in light blue.

Caracterización Pobreza

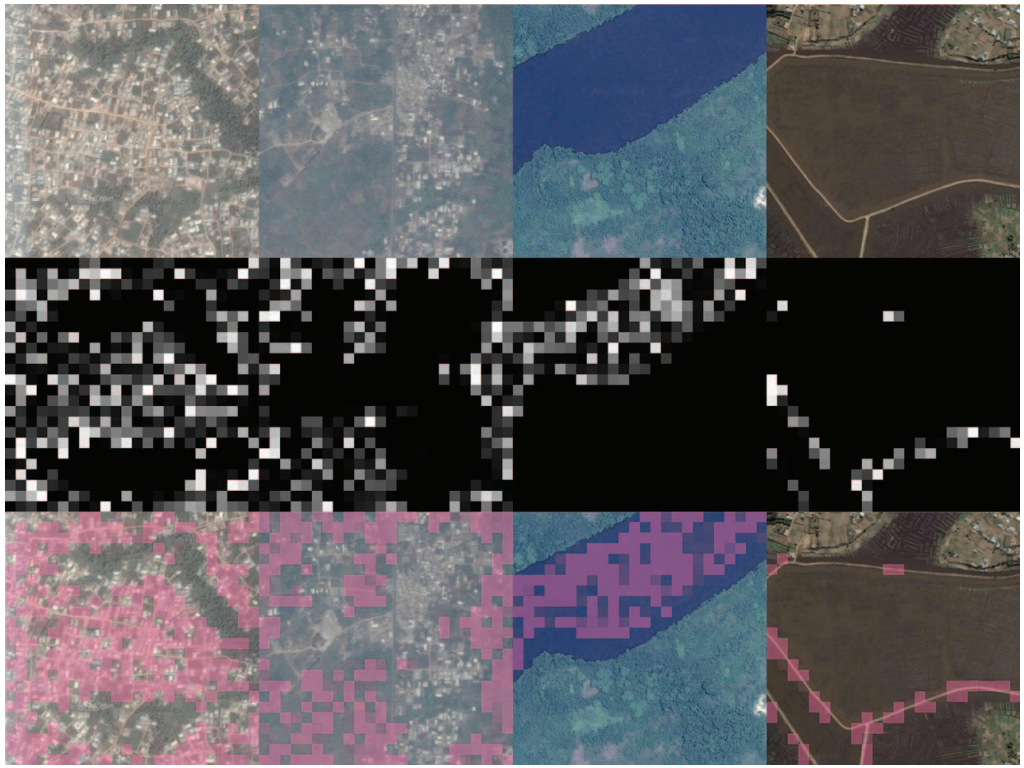


Fig. 2. Visualization of features. By column: Four different convolutional filters (which identify, from left to right, features corresponding to urban areas, nonurban areas, water, and roads) in the convolutional neural network model used for extracting features. Each filter "highlights" the parts of the image that activate it, shown in pink. By row: Original daytime satellite images from Google Static Maps, filter activation maps, and overlay of activation maps onto original images

Estimación Pobreza

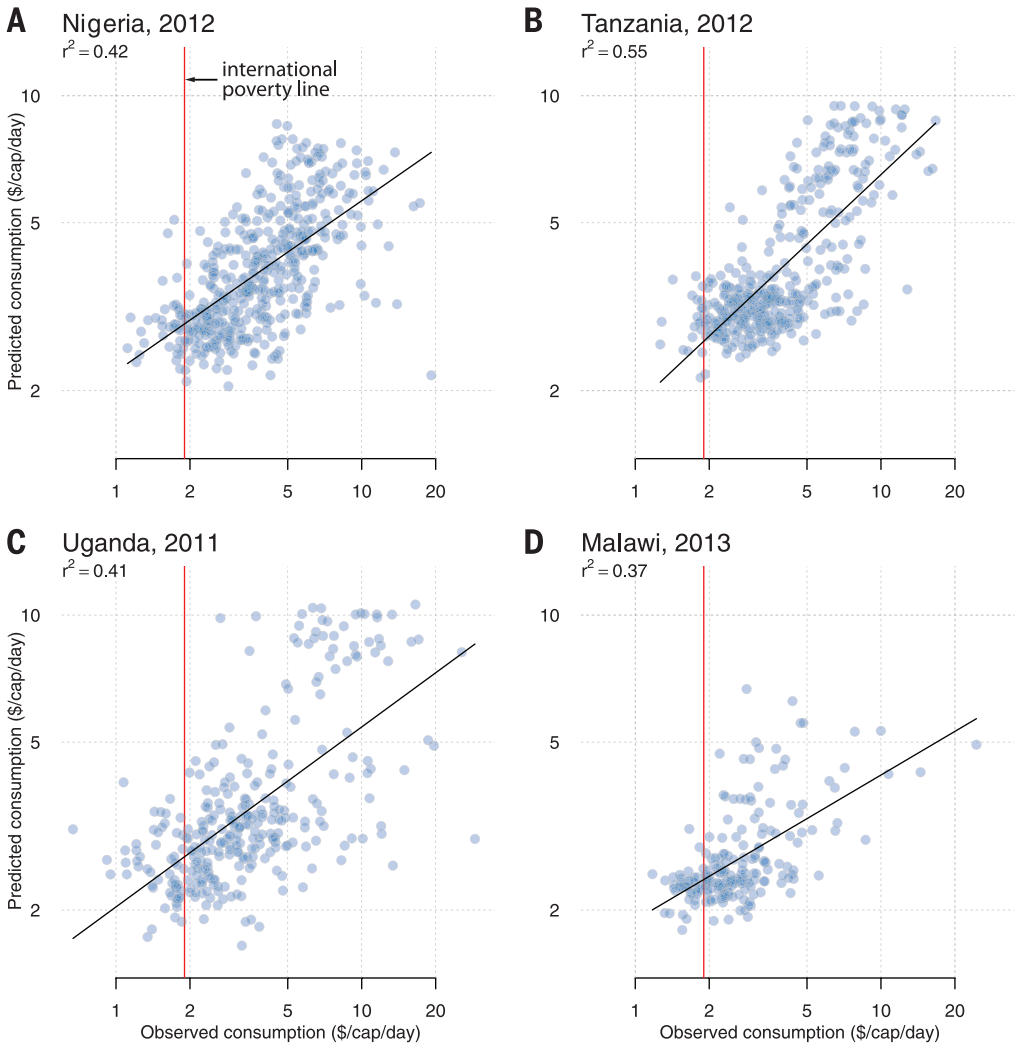


Fig. 3. Predicted cluster-level consumption from transfer learning approach (*y* axis) compared to survey-measured consumption (*x* axis). Results are shown for Nigeria (**A**), Tanzania (**B**), Uganda (**C**), and Malawi (**D**). Predictions and reported r^2 values in each panel are from fivefold cross-validation. Black line is the best fit line, and red line is international poverty line of $1.90 per person per day. Both axes are shown in logarithmic scale. Countries are ordered by population size.