

Detection of Phase Transition via Convolutional Neural Networks

Proyecto Final

Estudiante de la Lic. En física de la universidad de Guanajuato

Alvaro Gabriel de la Torre Navarro, 912472, ag.delatorrenavarro@ugto.mx

Introducción

En mecánica estadística es un trabajo usual el de caracterizar las fases de sistemas de estudio, sin embargo, hay ciertas fases como la topológica que no consta de parámetros de orden claros, u otros problemas que ni siquiera es posible medir aún. Las técnicas del aprendizaje de computadora (ML) basadas en CNN que se han estado desarrollando desde el record de ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), aplicado a investigar fases de materia con gran exactitud en clasificar fases de sistemas de 2D y de 3D. Sin embargo, estos métodos empleados requieren de información a priori que previene la aplicación de las técnicas de ML a sistemas desconocidos hasta ahora.

Objetivo

Crear una arquitectura de red neuronal convolucional (CNN) donde se muestre la posibilidad de la detección de la red de cambios de fase y la de hacer estimaciones cercanas a la temperatura crítica en sistemas de 2 dimensiones sin información a priori.

Justificación

En general es difícil o simplemente difuso el encontrar parámetros de orden; una de las formas en las que definimos las fases y sus transiciones, la caracterización de estos a veces ni es posible medirlas en experimentos. El aprendizaje de computadora no supervisado ha mostrado ya hasta cierto nivel, la posibilidad de encontrar estas fases, sus transiciones o incluso graficar diagramas de fase, sin la necesidad de información a priori proveniente de parámetros de orden o de la misma temperatura crítica, por lo tanto, eliminando las dificultades anteriores. El resolver el modelo de Ising de esta forma, con propuestas por la red cercanas a la real, no sólo indica la posibilidad de usarla como una herramienta correcta para este modelo, si no, también para aún más modelos, Ising es un modelo de juguete de benchmark y bien puede uno aplicar lo mismo a otros modelos. i.e. Potts, XY, Heisenberg.

Si bien ya se han usado propuestas de técnicas de ML como autoencoders para resolver problemas del tipo, no es evidente que las variables latentes nos provean de la temperatura crítica. Las CNN además, puede con los filtros hablar de relaciones espaciales escondidas en el conjunto de datos, hasta mostrar de forma cuantitativa la tendencia de los pesos a dilucidar la frontera de la transición. Obteniendo así visuales intuitivas.

Marco Teórico

La primer parte del aprendizaje por la red es la de la convolución, donde aprenderá las relaciones espaciales del sistema. Cada configuración o imagen tomada pasa por la red. Con activación ReLu para evitar la anulación del gradiente durante el aprendizaje.

$$\begin{aligned} & \{\sigma_{xy}\} \\ & \xrightarrow{\text{conv}} \sum_{i,j=1}^{N_f} \sigma_{(sX+i)(sY+j)} F_{ij}^a = \Sigma_{XY}^a \\ & \xrightarrow{\text{ReLU}} \max(0, \Sigma_{XY}^a) = u_{XY}^a \\ & \xrightarrow{\text{flatten}} \vec{u} = [u_{11}^1, u_{11}^2, \dots, u_{11}^C, u_{21}^1, u_{21}^2, \dots, u_{21}^C, \dots] = [u_m]. \end{aligned}$$

Luego sigue la parte de la red densa que toma el conjunto de mapas de características aplanado y regresa la probabilidad de que dada configuración pertenezca a un intervalo de temperatura.

$$\begin{aligned} & [u_m] \\ & \xrightarrow{\text{fully-connected}} \sum_{m=1}^{L^2/s^2 \times C} W_I^m u_m = z_I \\ & \xrightarrow{\text{Softmax}} \frac{e^{z_I}}{\sum_{J=1}^N e^{z_J}} = \beta_I^{\text{CNN}}. \end{aligned}$$

Por lo que se tienen 2 tipos de pesos en el sistema actualizables, esta es la llamada capacidad de la red. Y nosotros sólo tomamos los pesos de la red para ver de forma cuantitativa y cualitativa el aprendizaje de la temperatura crítica, sin embargo, los filtros también muestran casos especiales.

F_{ij}^a in convolution, W_I^m in fully connected layer.

$$\begin{bmatrix} a = 1, \dots, C \\ i, j = 1, \dots, N_f \end{bmatrix} \quad \begin{bmatrix} m = 1, \dots, L^2/s^2 \times C \\ I = 1, \dots, N \end{bmatrix}$$

La forma en la que se crearon los datos estocásticos de las configuraciones de magnetización del sistema se da así

$$\mathcal{T}_L = \left\{ (\{\sigma_{xy}^{(n)}\}, \vec{\rho}_n) \mid \frac{1}{\beta_n} = T_{\min} + n\delta \right\}_{n=0, \dots, (N_{\text{conf}}-1)},$$

Dónde delta es la definición o tamaño más pequeño entre intervalos de temperaturas.

$$\delta = (T_{\max} - T_{\min})/N_{\text{conf}}$$

La función de energía o Hamiltoniano que define como es que el algoritmo de Monte Carlo debe acercarse a la configuración de menor energía es

$$H = - \sum_{x,y} \sigma_{xy} (\sigma_{(x+1),y} + \sigma_{x,(y+1)}),$$

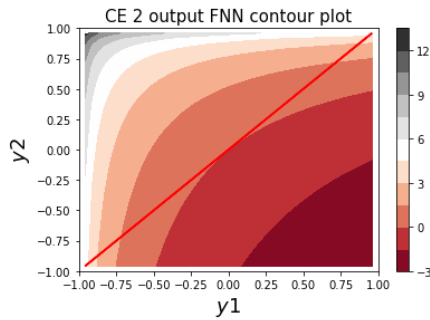
Usamos la representación one-hot de las temperaturas inversas para el sistema ya que no queremos que el sistema interprete que la sucesión intrínseca de baja a alta de las temperaturas se deba juntar al aprendizaje.

$$cl_N : \beta \rightarrow \vec{\beta} = \begin{cases} (1, 0, \dots, 0, 0) & \text{for } \beta < 0 \\ (0, 1, \dots, 0, 0) & \text{for } \beta \in \left[0, \frac{1}{N-2}\right) \\ \dots \\ (0, 0, \dots, 1, 0) & \text{for } \beta \in \left[\frac{N-3}{N-2}, 1\right) \\ (0, 0, \dots, 0, 1) & \text{for } 1 \leq \beta \end{cases}$$

La función de error que usamos es simplemente para poder evitar que se atrape uno en mínimos locales,

$$E(\vec{\beta}^{\text{CNN}}, \vec{\beta}) = - \sum_{l=1}^N \beta_l \log \beta_l^{\text{CNN}}$$

sin embargo, existe el problema de la sobre especialización



Y se debe tener cuidado a la hora de definir las épocas de entrenamiento.

Finalmente, con forme a la red, los autores proponen un parámetro de orden que sea dependiente del aprendizaje no supervisado de la red, (la suma va sobre los componentes de m , obteniendo un arreglo de tamaño N clases, donde l es el intervalo i -ésimo de las clases)

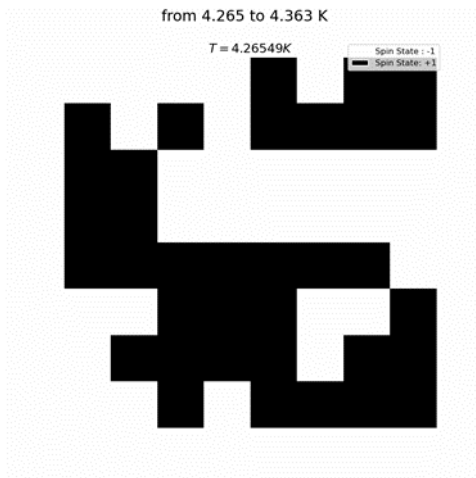
$$W_{\text{sum}}(\beta) = \sum_m W_l^m cl_N^l(\beta),$$

a la cual ajustamos con una función obtenida de la teoría de la aproximación de campo media motivada por la magnetización del modelo de Ising. Una de las constantes de ajuste, la parte que define el primer punto de inflexión de la función (β_{CNN}), es la propuesta de la red a la temperatura crítica.

$$a \tanh[c(\beta - \beta_{\text{CNN}})] - b,$$

Diagrama de bloques

El conjunto de datos usado es una malla de tamaño 8x8 que se extiende de temperaturas 0.1 a 5.0 donde cada cuadro representa el estado del spin de un átomo en un sistema completo, el sistema completo como el que sigue



Es la configuración más posible probabilística a esa temperatura, estos datos estocásticos se crearon con el uso de MCMC metrópolis, un total de 10000 diferentes conf. En el rango de temperaturas distanciados a intervalos iguales donde cada configuración se dejó correr por 10000 para encontrar la configuración con menor energía. Se hace el hincapié del uso de un rango de temperaturas grande para poder observar el comportamiento amplio del sistema. Sin embargo, para el problema de la detección de la transición de fase es innecesario las temperaturas negativas, debajo de 0.2 temperatura inversa y las mayores a 1. En la temperatura inversa. Se quitan por lo tanto estas partes del aprendizaje reduciendo así el tamaño de las 10000 configuraciones con las que empezamos, y reduciendo el número de clases a menos de 100. No son de usó principalmente por que las de baja temperatura tiene mucho orden de spin -1 por la aplicación de un campo magnético negativo débil para tener el estado fundamental único a temperatura cero.

La técnica de ML usada es la CNN, un modelo simple personalizado que consta de una capa de convolución, con su respectiva actuación de ReLu, procedimiento de aplanamiento, y una capa densa con actuación de Softmax.

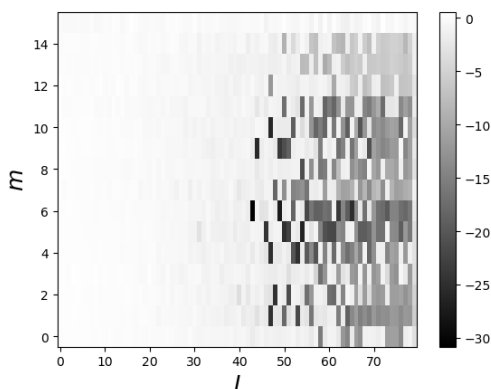
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 4, 4, 1)	10
flatten_1 (Flatten)	(None, 16)	0
dense_1 (Dense)	(None, 100)	1700
Total params: 1,710		
Trainable params: 1,710		
Non-trainable params: 0		

Se usó un optimizador de Adam bajo la exactitud, donde la perdida va sobre la etiqueta de nuestro grupo de datos prima en forma *one-hot*. La validación es en contra de 20% de los datos del grupo.

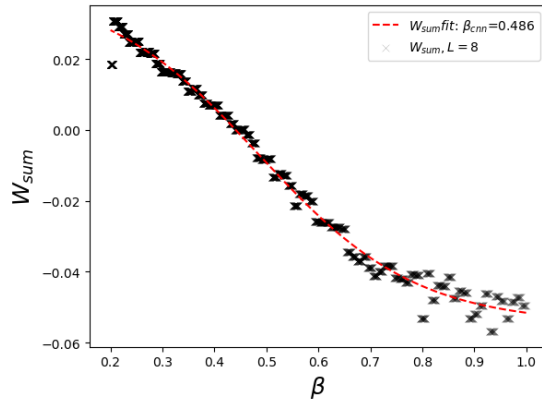
Se usa además herramientas de *Keras* como *EarlyStopping* y *ModelCheckpoint* para evitar la sobre especialización del entrenamiento; probablemente proveniente de la función de error, y pueda uno quedarse con el modelo que mejor validó en las épocas, también se deja un rango pequeño extra para explorar el espacio de modelos mejor ajustados.

```
es = EarlyStopping(monitor = 'loss', min_delta = 0.01, patience = 250, verbose = 1)  
model_cp = ModelCheckpoint(filepath = 'best_model{epoch}.h5', monitor = 'loss', save_best_only = True, verbose = 1, save
```

Se esperaba obtener una clara frontera de la representación gráfica de los pesos de la red densa sobre la temperatura crítica, esto ligado a uno de los 2 casos estadísticos posibles de signo de los filtros. Si positivo, caso de frontera inhomogéneo en al mapa de calor de los pesos, si negativo caso homogéneo en el mapa. Esto a 10000 épocas.



Y a menor cantidad de épocas, desde 8 a 12, se observa que podemos obtener de forma cuantitativa el valor de la temperatura crítica del sistema en transición, esto derivado del ajuste obtenido del parámetro de orden propuesto del CNN.



Actividades de programación

Se generó primero el grupo de datos utilizando *MCMC metrópolis* en un rango de temperaturas.

Se transformaron de forma lineal a una representación de *hot-one*. Se removió de segmentos innecesarios.

Se programo la arquitectura propuesta en el artículo y las visualizaciones correspondientes.

Se analizó la arquitectura y la posibilidad de mejora bajo el comportamiento de la perdida.

Se creo un parámetro de orden en el código para ajustarlo y obtener la propuesta de la red al problema.

Referencias

- [1] Tanaka A. and Tomiya A., Detection of Phase Transition via Convolutional Neural Networks, Journal of the Physical Society of Japan, 86, 6, 063001, 2017, doi:10.7566/JPSJ.86.063001, <https://doi.org/10.7566/JPSJ.86.063001>
- [2] Oh, Sang-Hoon. (2012). Contour Plots of Objective Functions for Feed-Forward Neural Networks. International Journal of Contents, 8(4), 30–35. <https://doi.org/10.5392/IJOC.2012.8.4.030>
- [3] Carrasquilla, J. (2016, May 5). *Machine learning phases of matter*. arXiv.org. <https://arxiv.org/abs/1605.01735>