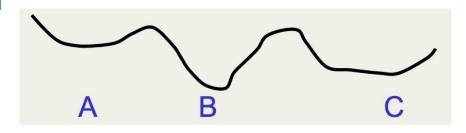


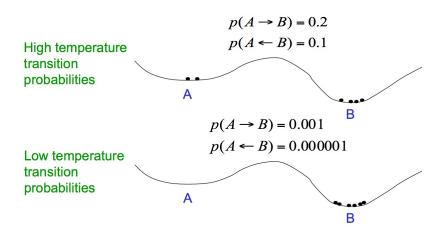
Problema de los mínimos locales en Hopfield



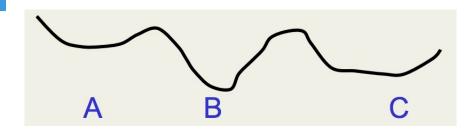
Los mínimos espúreos tienen más energía que las memorias verdaderas.

Una red de Hopfiled siempre toma decisiones que reducen la energía: esto hace que sea imposible escapar de mínimos locales.

Efectos de la temperatura en Hopfield estocástico



Simulated annealing



Simulated annealing es un método que nos sirve para escapar de mínimos locales.

Empiezo a una temperatura muy alta, y reduzco lentamente la temperatura.

Unidades estocásticas binarias

Toman decisiones random con cierto bias.

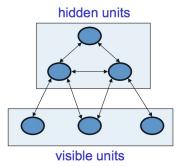
La temperatura controla el monto del ruido. Elevar el nivel del ruido es equivalente a decrementar los gaps de energía entre las configuraciones.

La función de activación será ahora una logística que depende de la temperatura:

$$p(s_i = 1) = \frac{1}{1 + e^{-\Delta E_i/T}}$$

donde $\Delta E_i = E(s_i = 0) - E(s_i = 1)$

Y si introducimos una capa oculta en Hopfield?



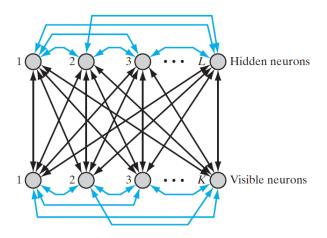
En vez de usar la red para almacenar memorias usarla para construir interpretaciones de la entrada:

La entrada está representada por las unidades visibles.

La interpretación estará representada por los estados de la las unidades ocultas.

La bondad de la interpretación es medida por la energía.

Boltzmann Machine



El objetivo de la Boltzmann machine es producir una red neuronal que modele correctamente la distribución de los patrones de entrada.

Hopfield vs. Boltzmann Machines

Similitudes:

Los estados de las neuronas son bipolares.

Los pesos son simétricos.

Las neuronas son seleccionadas en forma random para la actualización asincrónica

No hay conexiones de auto-feedback

Diferencias:

Las BM tienen una capa oculta.

La actualización de los estados en Hopfield es determinística, y en BM es estocástica.

En Hopfield los pesos se codifican al inicio. En BM los pesos se aprenden a través de una combinación compleja de simulated annealing y gradiente descendiente.

BM operando

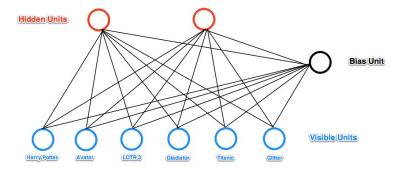
Given	A Boltzmann machine with weights specified in advance.
Initialize	$\hookrightarrow T_0, k = 0$ (temperature, iteration index)
	$\hookrightarrow T_{\min}, c$ (temperature limit and contraction)
	↔ Clamp the known signal values
	\hookrightarrow Randomize remaining signals to values in $\{-1, 1\}$
Iterate	○Repeat
	Operational Details and American Services
	○Repeat
	{
	\rightsquigarrow Select neuron I randomly.
	\rightsquigarrow If not a clamped neuron, given its signal, s_I
	\sim Compute: $x_I = \sum_{j} w_{jI} s_j$
	\sim Compute: $P = \frac{1}{1 + e^{-x_I/T}}$
	\rightsquigarrow if rand[0, 1) $< P$, set $s_I = 1$
	\rightarrow else set $s_I = -1$
	} until(all nodes are updated many times)
	\rightsquigarrow Reduce temperature: $T_{k+1} = cT_k$
	$\{ \text{until } (T_{k+1} < T_{\min}) \}$

Problemas de la BM

La máquina de Boltzmann, aunque teóricamente es un modelo con inmenso poder de cómputo, lamentablemente tiene serios problemas prácticos. Concretamente, deja de aprender correctamente cuando la máquina es escalada a algo más grande que una máquina trivial.

Tarea para el hogar: Resolver este problema :-)

Restricted Boltzmann Machines



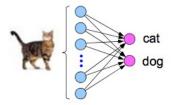
Se llaman restringidas, porque eliminamos las conexiones entre neuronas de una misma capa.

Restricted Boltzmann Machines

Objetivos

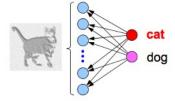
- 1. Aprender una representación abstracta del dataset de entrada sin supervisión (o sea, extraer features).
- 2. Una vez entrenado, reconstruir los datos que más probablemente se correspondan con esos rasgos.

Ejemplo utilización de RBM



Forward pass:

given these pixels, which hidden neuron should be activated?



Backward pass:

given a cat, which pixels distribution should be expected?

Restricted Boltzmann Machines: entrenamiento

Repetir hasta que el error entre el dataset de entrada y su reconstrucción caiga debajo de cierto umbral, o hasta alcanzar un número máximo de épocas:

- 1. Tomar una muestra v, computar las probabilidades de las neuronas ocultas, y muestrear un vector de activación h de esta distribución de probabilidad.
- 2. Computar el producto externo entre *v* y *h* y llamarlo gradiente positivo.
- 3. De h, reconstruir v^0 , y calcular h^0 de esto.
- 4. Computar el producto externo de v^0 y h^0 y llamarlo a esto el gradiente negativo.
- 5. Actualizar la matriz de pesos W con el gradiente positivo menos el gradiente negativo, multiplicado por el factor de aprendizaje: $\Delta W = (vh^T v^0h^{0T})$
- 6. Actualizar los bias a y b según: $\Delta a = (v v^0)$, $\Delta b = (h h^0)$

Laboratorio: What I cannot create, I do not understand. Richard P. Feynman

hat I commot create, Why count × soc I do not understand. Bethe Ansity P now how to robe every Keenler 3-0 Hall problem that has been robust werell. Temp Non Linear Vorment

Laboratorio

Misión 1: Hopfield estocástico

Programar una red de Hopfield estocástica.

Misión 2: Restricted Boltzmann Machines

Entender y hacerle modificaciones al código en:

https://github.com/echen/restricted-boltzmann-machines