

Balanceando Exploración y Explotación

Enrique Balp Straffon

enrique@synx.co

ebalpstraffon@synx.co



¿Cuándo, cómo y porqué explorar en machine learning?



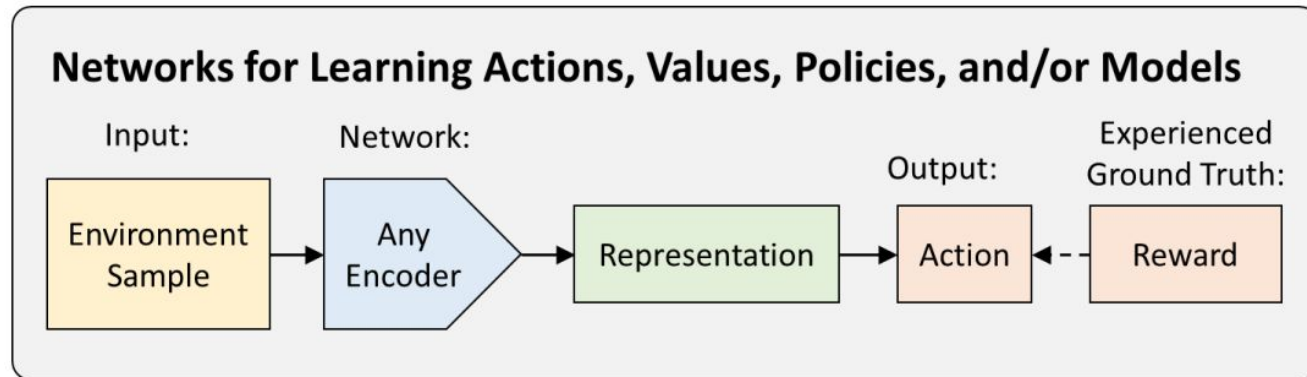
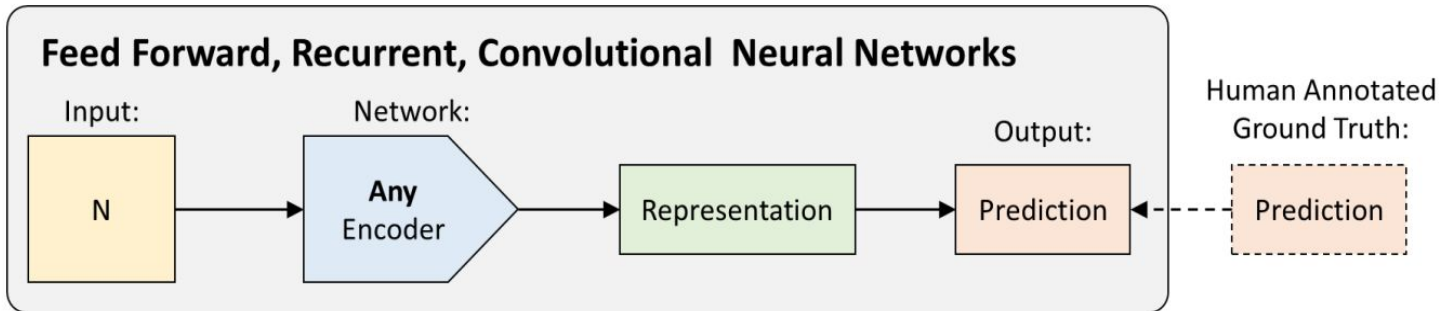
¿Cómo hacer que tu modelo de machine learning sepa lo que no sabe y explore lo que no conoce?



Contextual bayesian multi-armed bandits



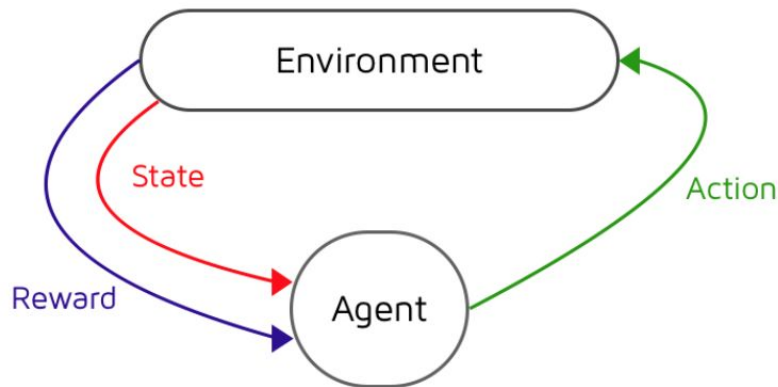
Supervised vs Reinforcement



Reinforcement Learning

At each step, the agent:

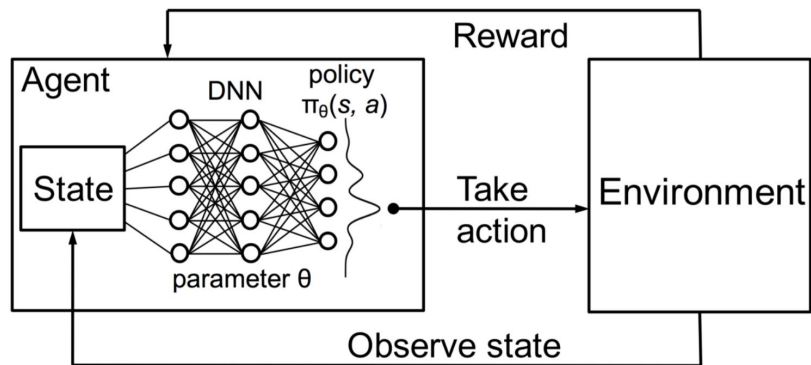
- Executes **action**
- Observe new **state**
- Receive **reward**

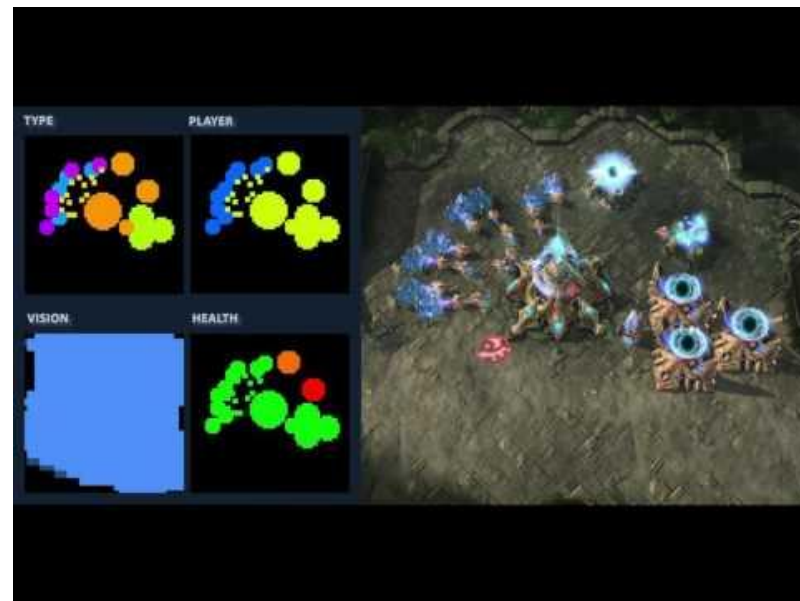
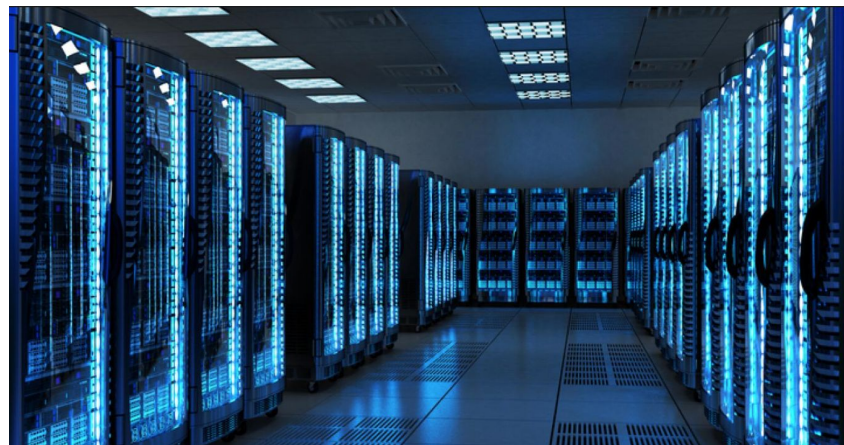


Exploration vs Exploitation

Playing Atari with Deep Reinforcement Learning

DeepMind Technologies,
2013





Unfortunately ... for practical applications ...

Deep reinforcement learning is surrounded by mountains and mountains of hype. And for good reasons! ...
Unfortunately, it doesn't really work yet.

Alex Irpan (Google Brain Robotics)

*“Often you can just approximate [your problem] as a **contextual bandits problem** ... There's a better theoretical understanding of contextual bandits problems.”- [John Schulman, OpenAI](#)*



action



reward

Multi-armed Bandit



state



action



reward

Contextual Bandit



state



action



reward

Full RL Problem

Multi-Armed Bandits

- Nos enfrentamos a N máquinas tragamonedas (bandits).
- Cada bandit tiene una probabilidad desconocida de distribuir una recompensa. Algunos bandits son muy generosos, otros no tanto.
- Al elegir un solo bandit por ronda, nuestra tarea es diseñar una estrategia para maximizar nuestras ganancias.

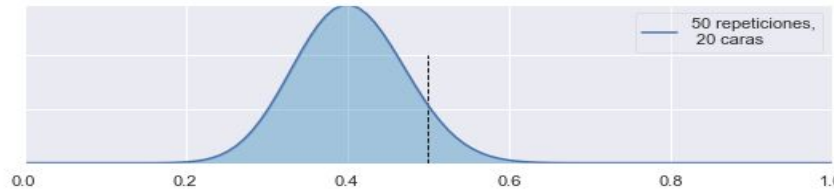
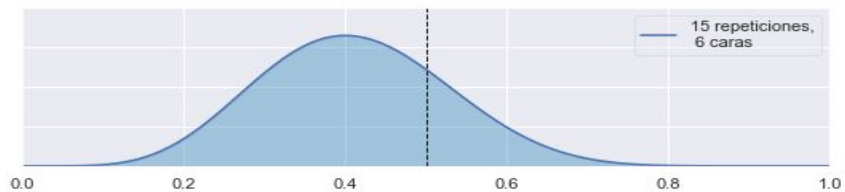
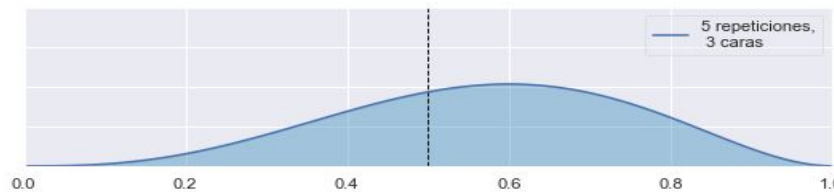
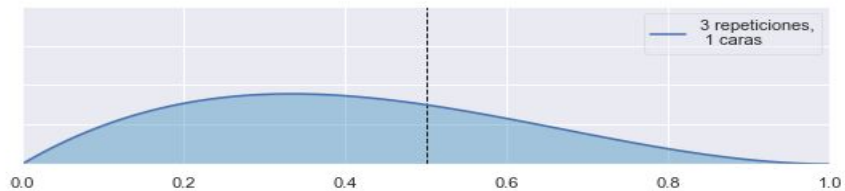
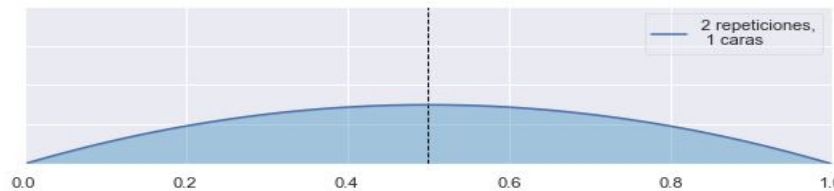
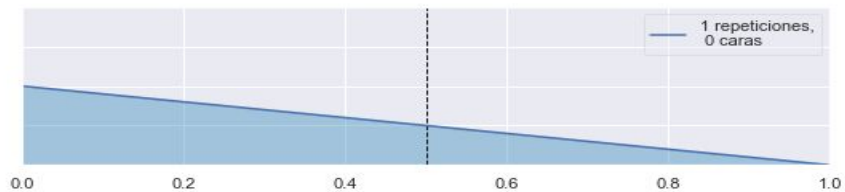


Exploration vs Exploitation

Interludio: Inferencia Bayesiana

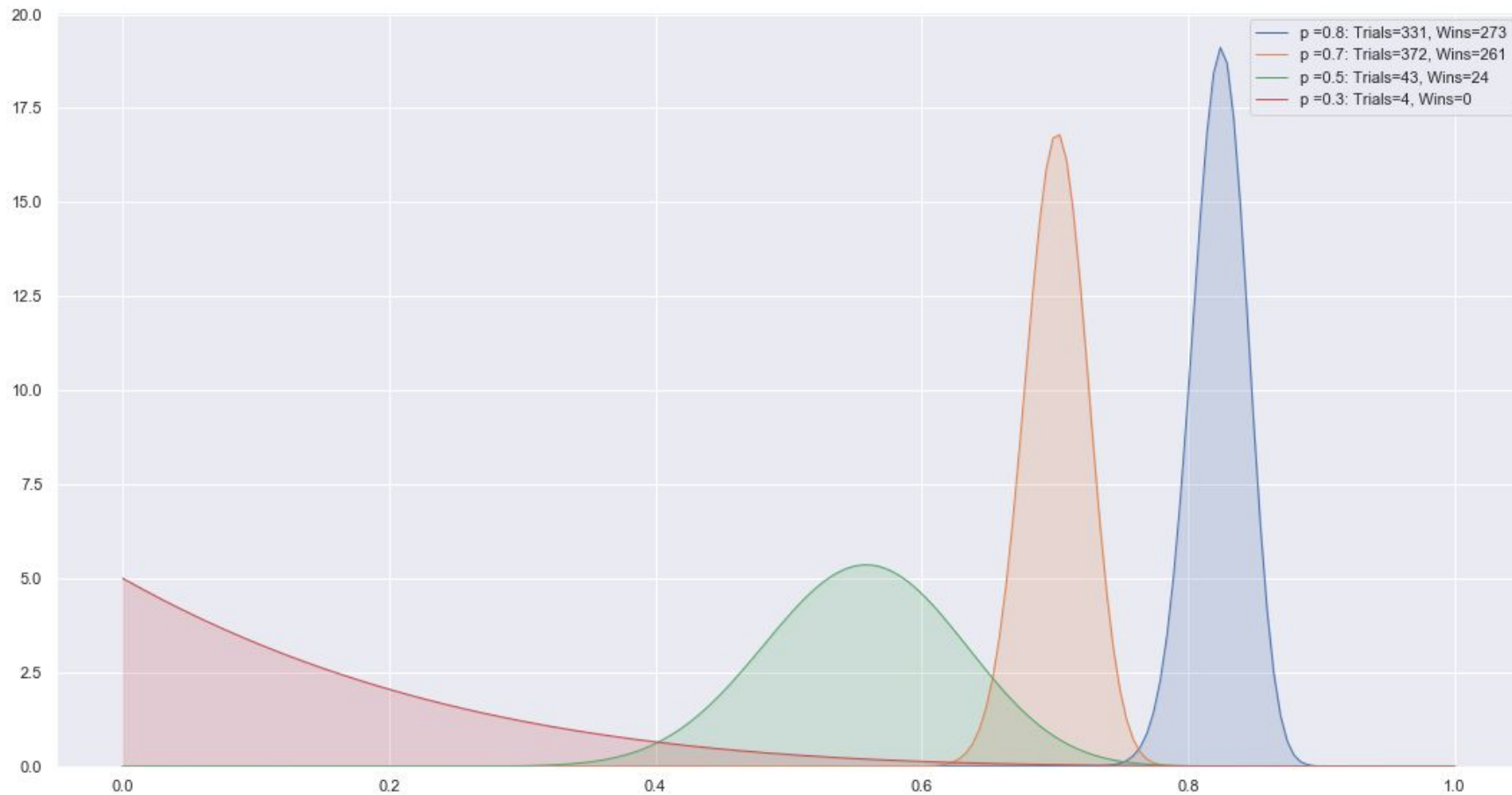
- En el enfoque **bayesiano**, la probabilidad se interpreta como una medida de **certeza**.
- La **inferencia bayesiana** se refiere al mecanismo para actualizar nuestra creencia al observar evidencia nueva.

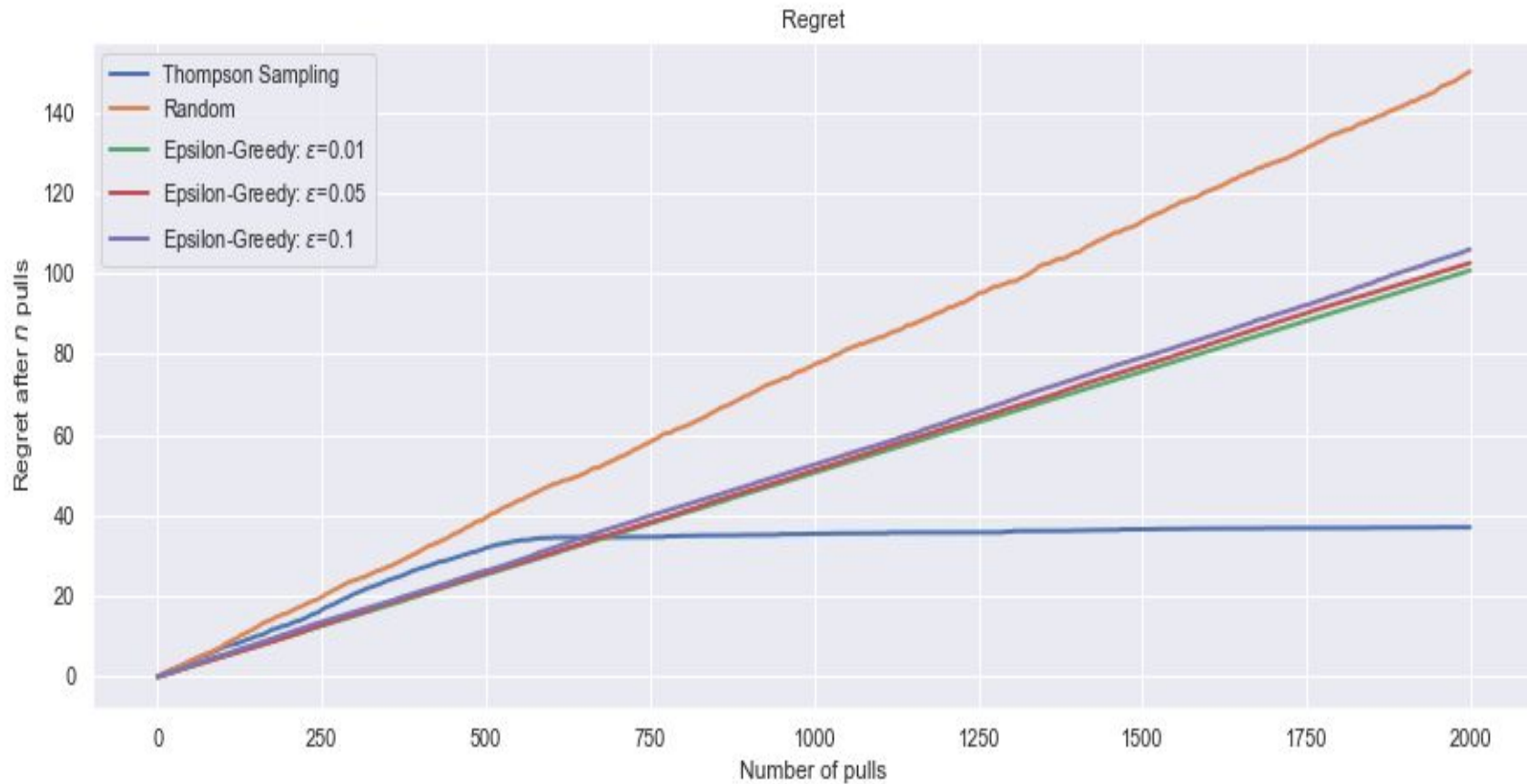
$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})} \propto P(\mathcal{D}|\theta)P(\theta)$$



Estrategia: Thompson Sampling

- Representamos cada bandit con una **distribución beta**, i.e. nuestra creencia sobre la probabilidad de distribuir recompensa.
- En la próxima ronda, muestreamos cada bandit (una única vez).
- Elegimos jugar con el bandit que obtuvo el valor máximo.





Contextual Bandits

(varios métodos)

- En cada turno el agente es presentado con información adicional sobre el **contexto** (o estado) del ambiente.
- El agente elige la **acción** tomando en cuenta esta nueva información.
- El valor de la **recompensa** también depende del contexto.



DEEP BAYESIAN BANDITS SHOWDOWN

AN EMPIRICAL COMPARISON OF BAYESIAN DEEP NETWORKS FOR THOMPSON SAMPLING

Carlos Riquelme*

Google Brain

rikel@google.com

George Tucker

Google Brain

gjt@google.com

Jasper Snoek

Google Brain

jsnoek@google.com

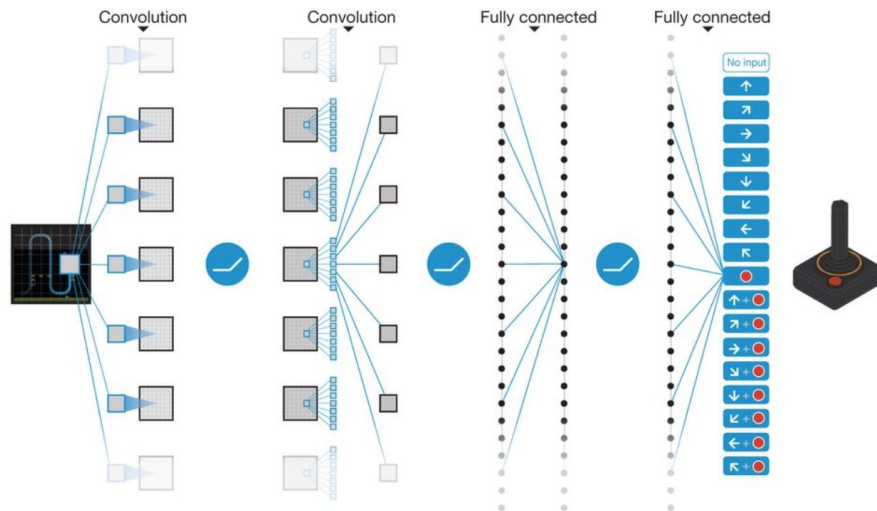
ABSTRACT

Recent advances in deep reinforcement learning have made significant strides in performance on applications such as Go and Atari games. However, developing practical methods to balance exploration and exploitation in complex domains remains largely unsolved. Thompson Sampling and its extension to reinforcement learning provide an elegant approach to exploration that only requires access to posterior samples of the model. At the same time, advances in approximate Bayesian methods have made posterior approximation for flexible neural network models practical. Thus, it is attractive to consider approximate Bayesian neural networks in a Thompson Sampling framework. To understand the impact of using

Neural Linear

- Entrenamos una **red neuronal**, la última capa tiene tantas neuronas como posibles acciones.
- La red neuronal trata de predecir las recompensas para cada acción.

Hacemos una regresión lineal bayesiana sobre la **última capa oculta**.



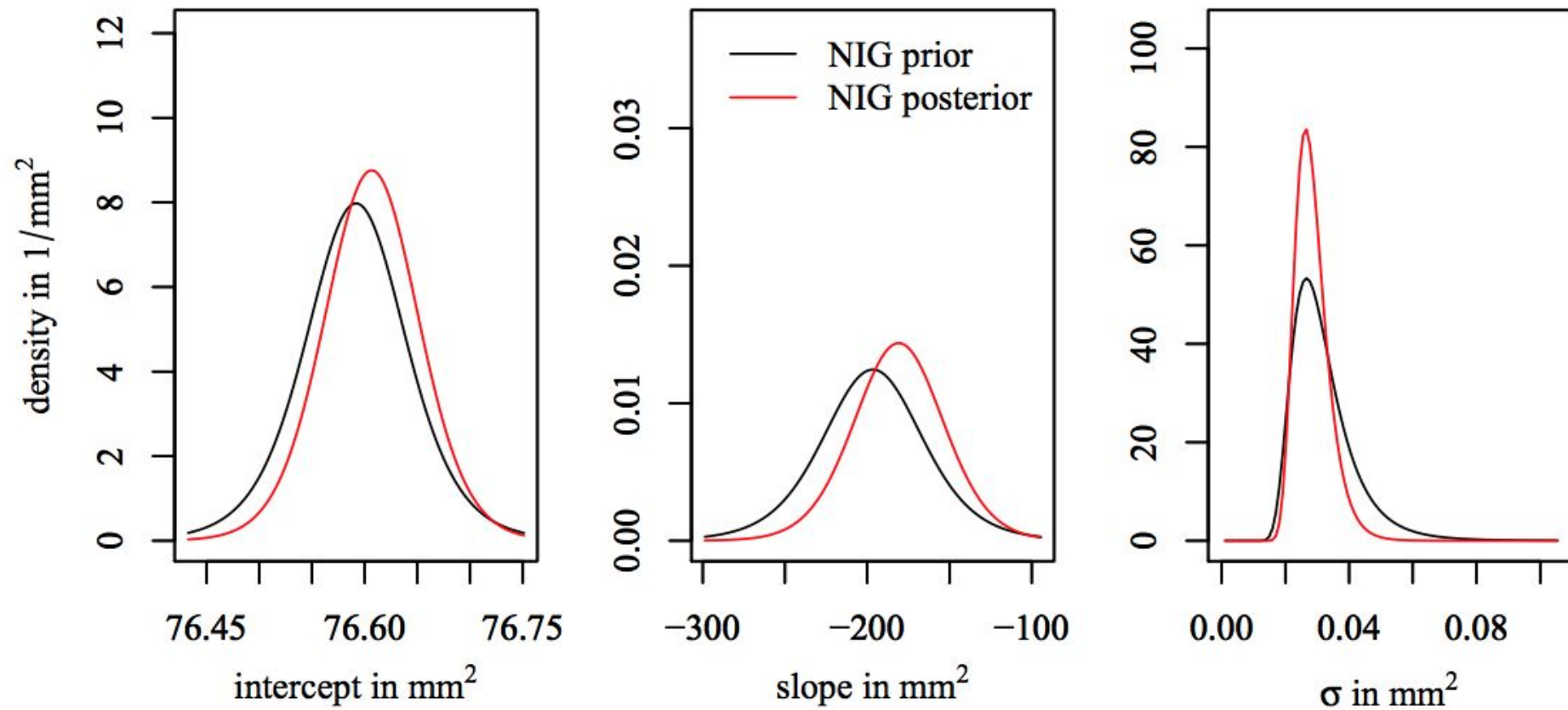
Regresión Lineal Bayesiana

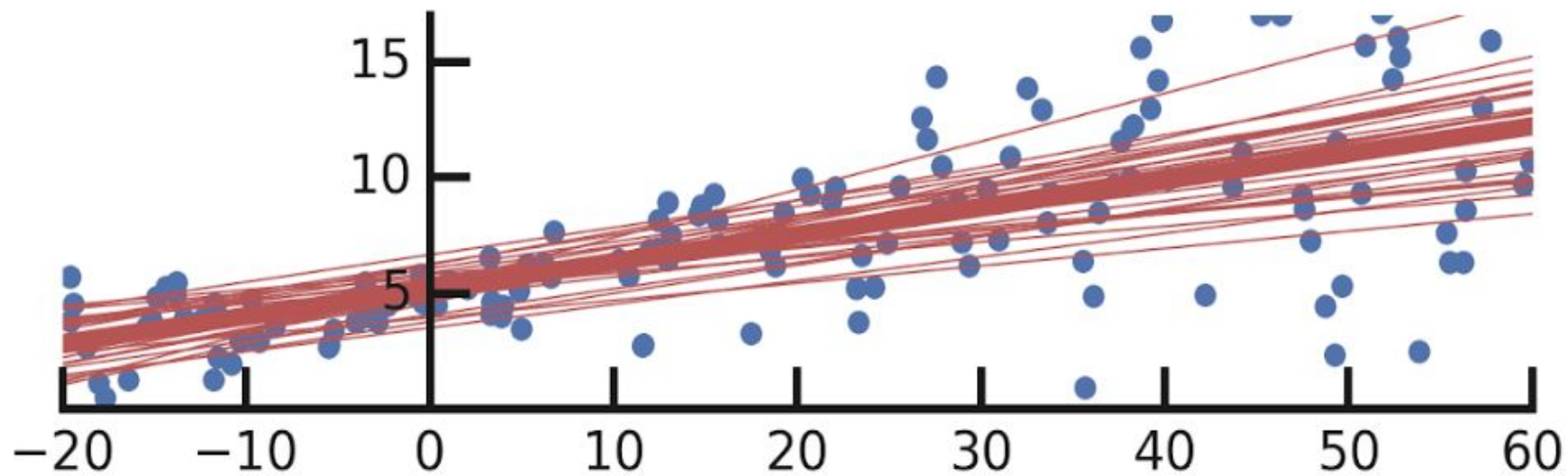
- En cada turno, ajustamos para cada Bandit, una Regresión Lineal Bayesiana a la recompensa Y con el contexto X como variable dependiente:

$$Y \sim X^T \cdot \beta + \epsilon \qquad \epsilon \sim N(0, \sigma^2)$$

- Modelamos la distribución conjunta de β y σ^2 para cada acción.
- Las actualizaciones de las distribuciones son calculadas analíticamente en cada turno.

Example






Thompson para Neural Linear

- En cada turno, muestreamos la distribución de β y σ^2 para cada bandit.
- Predecimos la recompensa con el modelo lineal, utilizando el contexto y los valores muestreados.
- Elegimos el bandit que maximiza nuestra predicción.
- Actualizamos los modelos de manera analítica (Bishop, 2006).

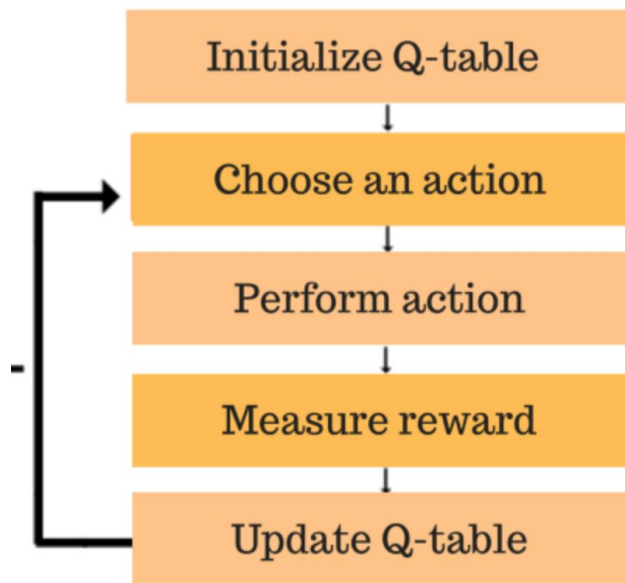
Q - Learning

$$Q^{\pi}(s_t, a_t) = \underline{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t, a_t]$$


Q-Values for the state
given a particular state

Expected discounted
cumulative reward

Given the state and action



$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$



↳ www.synx.co