Random Features for Large Scale Kernels (Caracterización aleatoria para kernels a gran escala)

- Sea $x \in \mathcal{X}$ y $\phi: \mathcal{X} \to \mathcal{H}$ una función de mapeo al espacio de Hilbert $\mathcal{H}$; tal que el producto interno: $K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$, con kernel $K: \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$; como función definida positiva.

- Sea $\{x_n \in \mathcal{X} \subseteq \mathbb{R}^P\}_{n=1}^N$; se estima la matriz $\mathbb{K} \in \mathbb{R}^{N \times N}$.
- Para problemas a gran escala $N \to \infty$.
- $\mathcal{H} \subseteq \mathbb{R}^Q$; $Q \to \infty$.
- Propuesta: Encontrar espacio de menor dimensión; $\mathcal{H} \subseteq \mathbb{R}^q$; $q < Q$; desde el mapeo aleatorizado $z: \mathbb{R}^P \to \mathbb{R}^q$, tal que:

$$K(x, x') = \langle \phi(x), \phi(x') \rangle \simeq \langle z(x), z(x') \rangle$$

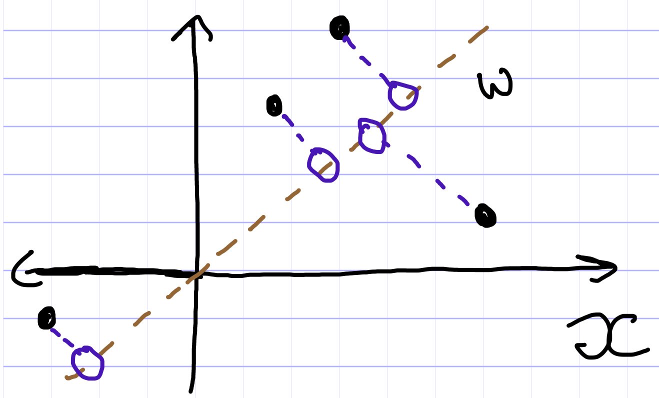
Aproximación de Kernel estacionarias (invariantes al desplazamiento)

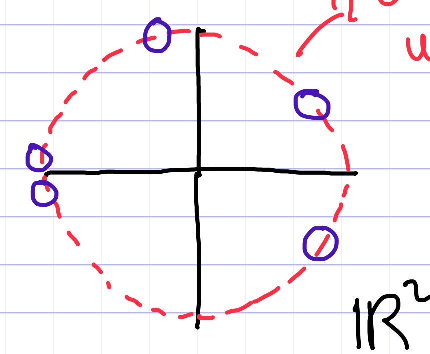
- Propiedad reproductiva:
  $$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

- Mediante aproximación aleatorizada:
  $$f(x) = \langle w, z(x) \rangle = w^T z(x)$$

Características de Fourier aleatorizadas.

- $\cos(w^T x + b)$; $w \in \mathbb{R}^q$; $b \in \mathbb{R}$.





Teorema Bochner. $k(x, x') = k(x - x')$ es definida positiva si y solo si $k(\delta)$ es la transformada de Fourier de una medida no-negativa.

- Si un Kernel estacionario $k(\delta)$ es escalado, Bochner garantiza que su transformada $p(w)$ es una función distribución.

- Sea $\xi_w(x) = e^{jw^Tx}$:

$$k(x-x') = F^{-1}\{p(w) | e^{jw^T(x-x')}\} = \frac{1}{2\pi}\int p(w) e^{jw^T(x-x')} \partial w$$

$$k(x-x') = \frac{1}{2\pi} \mathcal{E}_{p(w)}\{e^{jw^Tx} e^{-jw^Tx'}\} \propto \mathcal{E}_{p(w)}\{\xi_w(x)\,\xi_w^*(x')\}$$

- Dado $k(x-x'), p(w) \in \mathbb{R}$:

$$k(x-x') = \int p(w)\left[\cos(w^T(x-x')) + j\,\mathrm{Sen}(w^T(x-x'))\right]\partial w$$

$$k(x-x') = \int p(w)\cos(w^T(x-x'))\,\partial w + j\int p(w)\,\mathrm{Sen}(w^T(x-x'))\,\partial w$$

$$k(x-x') = \mathrm{Re}\{k(x-x')\} + j\,\underbrace{\mathrm{Im}\{k(x-x')\}}_{0}$$

Por ende se puede trabajar solo con cosenos.

$\rightarrow z_w(x) = \sqrt{2}\cos(w^T x + b)$; $\quad w \sim p(w)$

$b \sim \mathcal{U}(b \mid 0, 2\pi)$

$\hat{k}(x - x') = \mathbb{E}_w\{z_w(x) z_w(x')\}$.

$\rightarrow$ Para mejorar la varianza del estimador:

$$\{z_{w_j}(x)\}_{j=1}^{q} = \mathbb{z}(x)$$

$$k(x - x') = \langle \mathbb{z}(x), \mathbb{z}(x') \rangle = \frac{1}{q}\sum_{j=1}^{q} z_{w_j}(x) z_{w_j}(x')$$

Algoritmo: Capa Kernel $\rightarrow$ Random Fourier

1. Entrada: $k(x, x') = k(x - x') \Rightarrow$ kernel estacionario
2. Salida: $\mathbb{z}: \mathbb{R}^P \rightarrow \mathbb{R}^q$; $\langle \mathbb{z}(x), \mathbb{z}(x') \rangle \approx k(x - x')$
3. Calcular $p(w) = \frac{1}{2\pi}\int k(\delta) e^{-jw^T x} \partial\delta = \frac{1}{2\pi}\mathcal{F}\{k(\delta)\}$

   $w \sim p(w)$; $b \sim \mathcal{U}(b \mid 0, 2\pi)$
4. $\mathbb{z}(x) = \sqrt{\frac{2}{q}}[\cos(w_1^T x + b_1), \dots, \cos(w_q^T x + b_q)]$
5. Crear capa según $\mathbb{z}$ y entrenar por backpropagation.

Algunos casos de interés:

Kernel Gaussiano: $k(\Delta) = e^{-\frac{\|\Delta^2\|_2^2}{2\sigma^2}}$

$$p(w) = (2\pi)^{-\frac{p}{2}} e^{-\frac{\|w\|_2^2}{2\sigma^2}}$$

Kernel Laplaciano: $k(\Delta) = e^{-\|\Delta\|_1}$

$$p(w) = \prod_j \frac{1}{\pi(1+w_j^2)}$$

Cauchy $k(\Delta) = \prod_j \frac{2}{1+\Delta_j^2}$

$$p(w) = e^{-\|\Delta\|_1}$$