Effects of deception in social networks

Gerardo Iñiguez, Tzipe Govezensky, Robin Dunbar, Kimmo Kaski, Rafael A. Barrio

En este artículo consideran que la morfogénesis es uno de los fenómenos más impactantes de la auto-organización, la cual consiste en la capacidad de los organismos multicelulares de construir sus propios cuerpos a partir de una sola célula huevo, cuya progenie se auto-ensambla en anatomías muy complejas con varios órganos y tejidos exactamente de la misma forma cada vez. Estas células se comunican con las células vecinas para decidir la forma que tendrán los órganos, dónde crecerán, cómo se interconectan y dónde se detienen. Esta emergencia de organismos complejos se puede entender a partir de ciertas reglas sencillas y ciclos de retroalimentación homeostática. La evolución aprendió a explotar las leyes de la física y la computación para implementar el software morfogenéticos, altamente robusto, que corre en un hardware celular codificado con un genoma.

Es robusto porque resiste a perturbaciones, reparando daños mediante regeneración. La morfogénesis es un proceso adaptativo, al grado que un organismo, al ser partido a la mitad, puede incluso formar dos nuevos organismos. ¿Cómo es que el colectivo de las células sabe qué construir y cuándo parar? Las ciencias genómicas y la biología de células madre dan solo una respuesta parcial a esta pregunta.

El modelo que desarrollaron en este trabajo se trata de un autómata celular bidimensional continuo que tiene la virtud de que la regla de actualización es una función diferenciable de los estados de los vecinos de la célula, lo que permite el uso de funciones de pérdida y de optimización numérica basada en gradientes, herramientas propias de las redes neuronales, el aprendizaje profundo y la programación diferenciable. Esta regla de actualización es el análogo al genoma de un organismo vivo, y es compartida por todas las células del autómata y son diferenciadas por la información codificada en la señal de activación que reciben, emiten y guardan en sus vectores de estado.

La regla de actualización es reminiscente de una convolución en procesamiento de señales y programación diferenciable, la cual es capaz de aprender el comportamiento deseado y tiene las fases de:

1. Percepción: define lo que cada célula percibe de su alrededor, implementada mediante una convolución 3x3 con un kernel fijo, motivada por el hecho de que las células biológicas dependen solo de los gradientes químicos para guiar el desarrollo del organismo. Estos se llaman filtros de Sobel
2. Regla de actualización: cada célula aplica una serie de operaciones al vector de percepción, las cuales consisten en convoluciones 1x1 y no-linealidades ReLU. Esta regla de actualización es aprendida pero cada célula corre la misa regla.
3. Actualización estocástica de la célula: cada célula se actualiza de manera independiente de las otras en intervalos de tiempo aleatorios, esto se implementa con una máscara aleatoria por célula que actualiza los vectores.
4. Enmascaramiento de célula viva: se ajustan los canales de las células vacías a cero para que no participen en los cálculos ni lleven estados escondidos. Esto modela el proceso de crecimiento que comienza con una sola célula.

Experimento 1: Aprendiendo a crecer

Se entrena al AC a alcanzar una imagen objetivo después de un número aleatorio de actualizaciones. Se inicializa una red de ceros con solo una semilla en el centro y se aplica iterativamente la regla de actualización. La función de pérdida se puede optimizar de forma diferenciable con respecto a los parámetros de la regla mediante backpropagation-through-time, el método estándar para entrenar redes neuronales recurrentes. Una vez que la optimización converge, se corren simulaciones para ver como los ACs entrenados desarrollan patrones a partir de la semilla. El resultado es que los patrones son muy claros pero en tiempos largos se vuelven inestables.

Experimento 2: Lo que persiste, existe

El experimento 1 es inestable en analogía con los sistemas dinámicos, a los que cada célula corresponde, pues todas comparten la misma dinámica y se encuentran acopladas localmente entre sí. Al entrenar el modelo de actualización se ajusta esta dinámica. Evitar la inestabilidad consiste en hacer que el patrón objetivo sea un atractor. Una forma de lograr esto es aplicando la función de pérdida periódicamente, entrenando así al sistema mediante backpropagation en periodos de tiempo largos, pero requiere mucha memoria, así que mejor se usa una estrategia de *sample pool* de estados semilla, de los cuales se toma una muestra para el paso de entrenamiento, la cual se reemplaza con los estados de salida. La dinámica aleatoria del sistema provoca que el sistema termine en varios estados incompletos o incorrectos, pero al refinar la dinámica, el modelo se vuelve más robusto haciendo que las muestras sean más propensas a llegar al patrón objetivo.

Al usar los estados finales como puntos iniciales, el AC aprende a persistir o incluso mejorar un patrón ya formado, además de desarrollarlo a partir de la semilla, generando un atractor. La regla de la célula aprende entonces a estabilizar y refinar el patrón.

Experimento 3: Aprendiendo a regenerar

Al dañar el estado final de los ACs, se observa que desarrollan capacidades regenerativas sin ser entrenadas explícitamente para ello. Esto se debe a que el sistema acoplado de células fue entrenado para generar un atractor hacia una forma objetivo, pero también puede suceder que desarrollen crecimiento descontrolado, sobre-estabilidad que resulta en una falta de respuesta al daño, e incluso autodestrucción. Para mejorar la regeneración, se puede incrementar la cuenca de atracción. Los modelos que son dañados durante el entrenamiento son mucho más robustos.

Experimento 4: Rotando el campo perceptivo

Lo que percibe la célula de sus células vecinas se modela estimando los gradientes los canales de estado usando filtros de Sobel, análogo a que cada agente tenga 2 sensores que apuntan en direcciones ortogonales que pueden percibir los gradientes de concentración de ciertos químicos a lo largo del eje del sensor. Se pueden rotar estos sensores rotando los kernels de Sobel, i.e. rotando el eje a lo largo del cual el paso de percepción calcula los gradientes, produciendo versiones rotadas del patrón sin necesidad de volver a entrenar a los ACs.

Esto es sorprendente pues rotar gráficos basados en pixeles involucra calcular un mapeo que no es necesariamente biyectivo y que clásicamente involucra interpolar entre pixeles, esto es porque un pixel probablemente se traslapará con otros al ser rotado. El desarrollo exitoso de patrones sugiere que hay una cierta robustez en las condiciones subyacentes ajenas a las experimentadas en el proceso de entrenamiento.

**Modelado Embriogenético**

Se ha encontrado que la morfología objetivo de organismos vivos no está totalmente codificada por el ADN, sino que se mantiene mediante un circuito psicológico que guarda un punto fijo para esta homeostasis anatómica. Existen algunas técnicas para reescribir este punto fijo, lo que resulta en que, por ejemplo, gusanos planos de 2 cabezas, después de ser cortados en pedazos en el agua corriente, generan más gusanos de 2 cabezas.

**Ingeniería y Machine Learning**

El método descrito en el artículo provee una forma de programar a un sistema de computadoras en un arreglo de células para que llegue a un estado global predefinido y para que recupere ese estado en caso de fallas por diversos elementos y reinicios, por lo que este modelo se puede usar para diseñar agentes autoorganizados y confiables. También se propone explorar un modelado de aprendizaje descentralizado basado en este método, en oposición al tradicional modelado global en el campo del aprendizaje profundo.