Información sobre el repositorio DeepElaticity

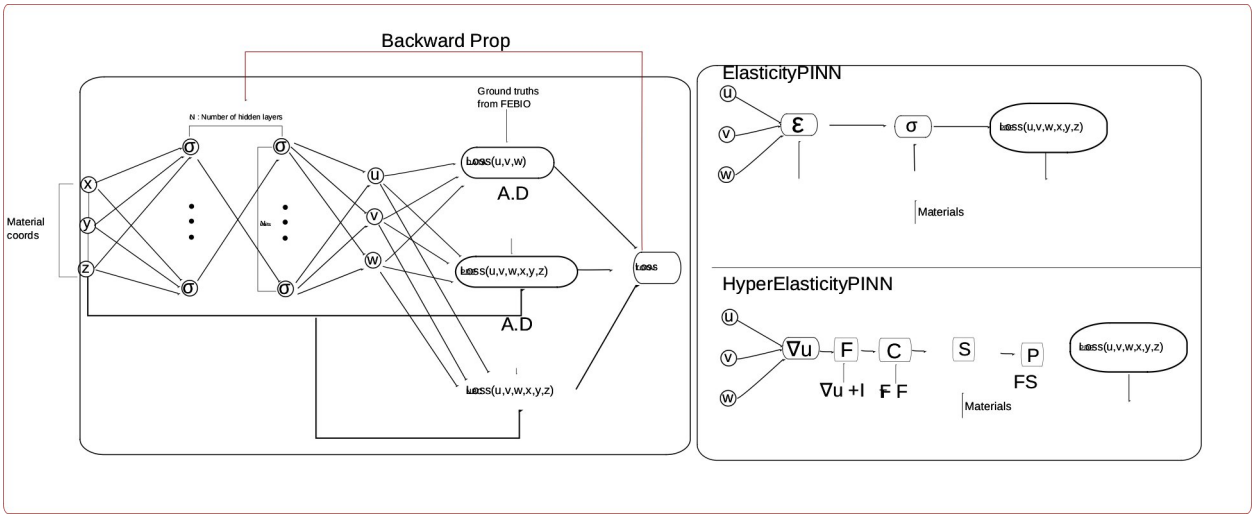
30 mayo 2024

Introducción

Este repositorio tiene todo el trabajo realizado sobre PINNs aplicadas a elasticidad.

El objetivo es el estudio de la aplicación de este tipo de redes sobre la deformación de tejido blando. La modelización de este tejido ha pasado de plantearse como Lineal, a Hiperelástico, y también se consideró el estudio de deformación por contacto (tema del que hay algunos artículos en la carpeta de *papers*).

El desarrollo de todo el código comenzó en el repo de https://github.com/CDiazCuadro/ElasticityPINN , en la rama de linear_prostata. Lo que ahí se hizo se puede encontrar en la carpeta de "*legacy_previo_repo*", además de algunos diagramas sobre el funcionamiento del código:



El repo se estructura en notebooks que es donde se realizaban las pruebas y experimentos; data que eran los datos utilizados; src que es para reutilizar código (source): informes que son reportes creados sobre el proyecto; docs que son notas y apuntes sobre la teoría, en su formato md y pdf, además de artículos científicos considerados; scripts, que son código desarrollado en notebooks pero que se dejaba ejecutando de noche también.

Nota: Para hiperelasticidad, los scripts y resultados están en la misma carpeta que los notebooks (esto fue por sencillez a la hora de experimentar, y porque había poco código).

Vamos a ir uno a uno explicando que contiene cada parte, pero antes una aclaración sobre el ordenador con gráfica A2.

Ordenador

El ordenador que contiene la gráfica potente con la que se están ejecutando las pruebas, tiene una series de usuarios, el usuario principal es “zeus”, cuya contraseña es “zeus1234”. Con este usuario se podrían crear más. El usuario que yo he estado usando “arturosf” y la contraseña es “arturosf1234”. Sin embargo para entrar via ssh, recomiendo poner las claves publicas de nuestro ordenador en el fichero de authorized_keys. El host ip es “158.42.20.146”, pero hay que estar dentro de la VPN de la universidad para que nuestro ordenador pueda accederlo.

```
Host zeus
    HostName 158.42.20.146
    User arturosf
```

Todo el desarrollo que se ha hecho ha sido con pytorch>=2.1 y python 3.10. Para la gestión del entorno se ha usado Anaconda, y el “requirements.txt” se puede encontrar en el repo (aunque no haría falta instalarlo todo al pie de la letra, la mayoría de versiones no tendrán conflictividad). Respeto al tema de los drivers, ese ha sido un tema que ha ocupado muchísimo tiempo, porque el ordenador tiene dos gráficas, una que esta en la placa base y se usa directamente para mostrar la interfaz gráfica por hdmi etc... Y la otra, que es la potente A2 de NVIDIA. Para esa segunda se necesitaban drivers de las versiones 535. Mientras que para la otra debíamos ir a versiones más antiguas. Tener dos drivers de nvidia a la vez no es posible por lo general, e instalar el driver antiguo evitaba que usaramos la gráfica nueva y viceversa. De modo que opté por instalar el driver moderno y usar la gráfica moderna pero esto limitaba la interfaz gráfica, de modo que solo he accedido al ordenador por medio de SSH. Para ello recomiendo usar Visual Studio con la extension de Remote Explorer, hace todo mucho más sencillo. Para ver la versión de la gráfica y su uso podemos usar los comandos “smi-nvidia” y “nvtop”.

Para ejecutar scripts y dejarlos ejecutando en el ordenador sin necesidad de tener el terminar abierto, recomiendo usar “nohup” o “screen” .

Data

La sección de data se divide en lineal e hiperelástica. Estos datos han sido proporcionados por María José, y tenemos varios ficheros clave.

Para el caso elástico lineal, se planteó el problema inverso, de modo que se aspiraba a inferir los parámetros materiales por medio de deformaciones conocidas. La primera carpeta la de “ARTURO_TEST_1” contiene información de nodos, elementos, fuerzas etc que será luego usada por todo el código, porque es troncal a todo el resto de datos (solo cambiarán las tensiones y desplazamientos). Sin embargo los desplazamientos en esta carpeta no son suponiendo la aproximación de pequeños desplazamientos. En la siguiente carpeta sí se supone esta aproximación “LINEAR_SMALL_DISP”. Y para cada paso de carga tenemos los desplazamientos u, v, w, además de las 6 componentes independientes de tensión.
Las otras carpetas (“MULTIPLE_E_VALUES” y “MULTIPLE_E_VALUES_NEW”) son lo mismo

pero para diferentes valores de los parámetros materiales (pues nos centrábamos en el problema inverso).

Para el caso hiperelástico (carpeta “002-Neo”), tenemos dos carpeta, la primera “ARTURO_TEST_2_NEO” tiene datos bajo el modelo de Neohook pero las tensiones están en formato “S1, S2, S3, SINT, SEQV ” y en la carpeta “DATOS_HIPERELASTICO_3”, tenemos los datos de tensiones con sus 6 componentes independientes, y para diferentes valores de E. En principio trabajaremos con los parámetros de lame para el caso de neohook, pero partimos de diferentes valores de E (los usamos en MPa), y un mismo valor de nu=0.4 .

Notebooks

Esto también esta separado en el caso lineal y en el hiperelástico.

Para el caso linear, en los dos primeros notebooks “000” se repasa lo que se hizo en el otro repositorio, y se reproducen resultados. En el siguiente notebook “001” se plantea un modelo mixto en que el output del modelo no es desplazamiento, sino desplazamiento y tensiones, algo mencionado en el artículo *“Solving Forward and Inverse Problems of Contact Mechanics using Physics-Informed Neural Networks”.*

En “002” y “003” se prueban otros tipos de PINNs o NNs.

En “004” se desarrolla ese código reutilizable que se encuentra en la carpeta “src”.

En “005” se obtienen los resultados que vemos en el informe justificativo de 2023.

En “006” se limitan los collocation points solo al área cubierta por la próstata.

En “007” se planteaba hacer una simulación de la deformación solo con condiciones de contorno e imponiendo la física, para verificar que la PINN está aprendiendo lo que debe.

Para el caso Hiperelástico, carpeta “002-HyperElasticity”, tenemos el notebook “001” donde planteo el caso hiperelástico, y luego en “001.1” se desarrolla el código para un loop recorra múltiples valores y configuraciones de la red. Eso se hizo ejecutando el script de .py

La carpeta resultados tiene muchas imágenes generadas durante el loop, además de han guardado los objetos pickle, que contienen toda la información del entrenamiento y el modelo propio.

Scripts

Aquí tenemos los ficheros .py que dejaba ejecutando, en lugar de dejar ejecutando notebooks, pero no tienen información ni código nuevo. Corresponden todos al caso elástico lineal.

Carpeta src

Esta carpeta tiene código reutilizable para cargar datos y modelos.

En la subcarpeta “utils” tenemos el módulo de la carga de datos, muy importante para pasar los datos a memoria y hacer una carga eficiente de ellos y separación en los diferentes conjuntos que se necesitan: datos de BC, datos de DATA, datos de PDE, collocation points.

En train.py tenemos un objeto “trainer” encargado del entrenamiento de los modelos que tenemos en el módulo “models.py”. En este último tenemos los modelos del caso linea, del casi hiperelástico, y algunas pruebas como la forma mixta de la que se hablaba más arriba.